The Resource Journal for IBM Micro Programmers

PROGRAMMER'S Journal

JANUARY/FEBRUARY 1988
VOLUME 6.1
$5.00

5TH YEAR ANNIVERSARY ISSUE

Porting Mainframe Software to the PC

The Software Approach and a Hardware Alternative
by
M. Steven Baker

FORTRAN

GRAPHICS

Write Mode 3 of the VGA
by
Michael Abrash

EMS

The LIM 4.0 Expanded Memory Specification
by
John J. Newlin

Routines that Access the Environment
Turbo Pascal 4.0
by
Tom Swan

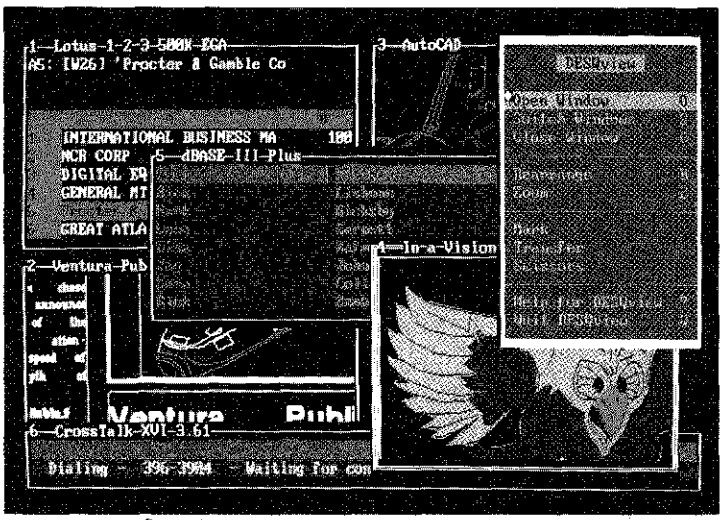
Pascal

A tasty treat...
SPECIAL 5 YEAR INDEX OF ARTICLES
see page 71

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

How to tell the difference between DESQview™ 2.0 and any other environment.

Selecting DESQview, the environment of choice, can give you the productivity and power you crave, without the loss of your old programs and hardware. If you like your existing programs, want to use them together, transfer data between them, print, sort, communicate with or process-in-background, yet still have the need to keep in place your favorite PC(8088, 8086, 80286 or 80386), DESQview is the "proven true" multitasking, multi-windowing environment for you. Best of all, DESQview 2.0 is here now, with all the money saving, time saving, and productivity features that others can only promise for the all-too-distant future.



One picture is worth a thousand promises.

And with DESQview's new graphics enhancements for Hercules, CGA, EGA, and VGA, Version 2.0 still offers the same award winning and pioneering features for programs that earned DESQview its leadership, only now you can also run desktop publishing programs, CAD programs, even GEM-™, Topview-™, and Microsoft Windows-™ specific programs. In some cases you'll add as little as 10–40K to your system overhead. Now you can have multi-tasking, multi-windowing, break the 640K habit too and still get an auto dialer, macros, menus for DOS and, for advanced users, a new complete application programmer's interface capability. No wonder that over the years, and especially in recent months, DESQview, and now DESQview 2.0 have earned extravagant praise from some of the most respected magazines in the industry.

"Product of the Year" by readers vote in InfoWorld.

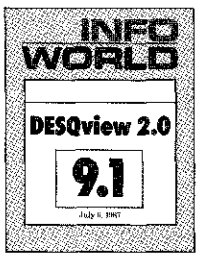
"Best PC Environment" by popular vote at Comdex Fall in PC Tech Journal's "System Builder" Contest.

"—I wouldn't want to run an IBM or compatible computer without DESQview"—InfoWorld, Michael Miller.

"A colossus among windowing environments"... "will run almost anything"—PC Week, Marvin Bryan.

"Windows, promises, but DESQview delivers"—MICRO-TIMES, Birrell Walsh.

No other environment has consistently pioneered features, openness, and productivity. See for yourself. Send in the coupon. The possibilities are endless with DESQview 2.0.



Attention Programmers: For more information about Quarterdeck's API, and future 386 program extensions, call us today.

SYSTEM REQUIREMENTS
IBM Personal Computer and 100% compatibles (with 8086, 8088, 80286 or 80386 processors) with monochrome or color display; IBM Personal System/2 • Memory: 640K recommended; for DESQview itself 0-145K • Expanded Memory (Optional): expanded memory boards compatible with the Intel AboveBoard; enhanced expanded memory boards compatible with the AST RAMpage • Disk: Two diskette drives or one diskette drive and a hard disk • Graphics Card (Optional): Hercules, IBM Color/Graphics (CGA), IBM Enhanced Graphics (EGA), IBM Personal System/2 Advanced Graphics (VGA) • Mouse (Optional): Mouse Systems, Microsoft and compatibles • Modem for Auto-Dialer (Optional): Hayes or Compatible • Operating System: PC-DOS 2.0-3.3; MS-DOS2.0-3.2 • Software: Most PC-DOS and MS-DOS application programs; programs specific to TopView 1.1, GEM 1.1 and Microsoft Windows 1.03 • Media: DESQview 2.0 is available on either 5¼" or 3½" floppy diskettes

Rush me DESQview 2.0! Today! PGMR-1/2

No. of Copies	Media 3½"/5¼"	Product	Retail Price ea.	Total
		DESQview 2.0	$129.95	$
		Shipping & Handling USA	$ 5.00	
		Outside USA	$ 10.00	$
		Sales Tax (CA residents)	6.5%	$
			Amount Enclosed	$

Payment: ☐ Visa ☐ MC ☐ AMEX ☐ Check

Credit Card: Valid Since ____/____ Expiration ____/____

Card Number:

Credit Card Name

Shipping Address

City State Zip Telephone

Mail to: Quarterdeck Office Systems, 150 Pico Boulevard, Santa Monica, CA 90405.
NOTE: If you own DESQview call us for a special upgrade offer, or send in your DESQview registration card. AST Special Edition users included.

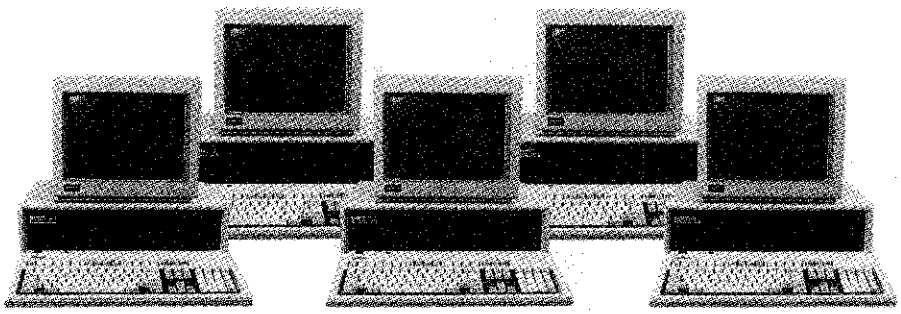
DESQview™
Quarterdeck

Quarterdeck Office Systems • 150 Pico Boulevard, Santa Monica, CA 90405 • (213) 392-9851

DESQview is a trademark of Quarterdeck Office Systems. AboveBoard is a trademark of Intel Corporation. Hayes is a trademark of Hayes MicroComputer Products Inc. IBM, PC, Personal System/2 and TopView are trademarks of International Business Machines Corporation. Microsoft Windows and MS are registered trademarks of Microsoft Corporation. Mouse Systems is a trademark of Metagraphics/Mouse Systems. RAMpage is a trademark of AST Research, Inc. GEM is a trademark of Digital Research. Hercules is a trademark of Hercules.

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

VM/386



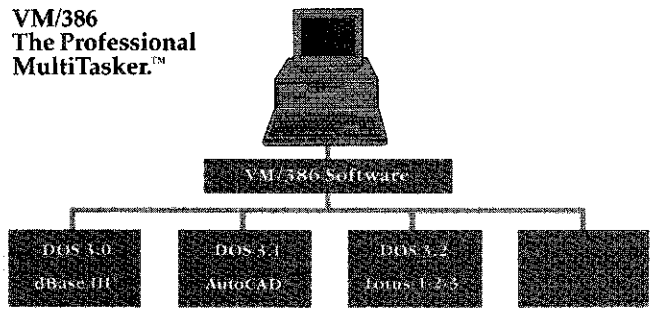
IT'S LIKE HAVING A DESKTOP FULL OF PCs

We know why you have it. Power. And now you can multiply that 386 power. Through true multitasking. With VM/386.™

VM/386 is the 80386 control program that brings you true DOS multitasking. VM/386 uses the virtual 8086 mode, built into the 80386 processor, to create individual virtual machines. You can load a different application in each virtual machine. It's like having a desktop full of PCs.

You have complete control over the virtual machines. You can tailor each virtual machine to fit your needs—and priorities. Each virtual machine has its own DOS, CONFIG.SYS, AUTOEXEC.BAT, and memory-resident programs along with its application. And each virtual machine is isolated from the others. A malfunction in one program doesn't destroy the others.

VM/386
The Professional MultiTasker.™



There is virtually no limit to the number, or kind, of applications that can run simultaneously. For example, you can recalculate a 1-2-3® spreadsheet, sort a dBASE III® file, and receive your E-mail—all at the same time. You can even run more than one AutoCAD™ program simultaneously. And EGA applications run in the background as well as in the foreground—perfectly.

VM/386 protects your investment in software—and hardware. VM/386 will run all DOS software. Existing and future. Without modification. VM/386 is compatible with 80386-based PCs—either native or Intel® Inboard 386.™ VM/386 will support the monitors, hard disks, printers, mice, modems—even the network—you're currently using—or plan to use.

With VM/386 you use familiar commands. You don't have to buy new software or upgrades to get true multitasking. No PIF files. No special loaders. VM/386 works the way you do. Only lots faster.

VM/386 is easy to install. Easy to learn. And easy to use.

Dramatically increase your productivity. Start using all the power built into your 80386. Phone or write today.

IGC
4800 Great America Parkway
Santa Clara, California 95054
(408) 986-8373

System Requirements

80386-based computer such as COMPAQ® DESKPRO® 386® or 80286 computer with Intel® Inboard™ 386

One 1.2 Mb (5¼") disk drive or, one 3½" microfloppy

One hard disk drive

DOS 3.0 or later

2 Mb memory recommended

Not copy protected

Package includes both 5¼" and 3½" media.

VM/386 is a trademark of IGC.
AutoCAD is a trademark of Autodesk Inc.
COMPAQ and COMPAQ DESKPRO 386 are registered trademarks of Compaq Computer Corporation.
dBASE III is a registered trademark of Ashton-Tate Corporation.
Intel is a registered trademark of Intel Corporation.
Inboard is a trademark of Intel Corporation.
1-2-3 is a registered trademark of Lotus Development Corporation.

Number **169** on the Reader Service Card

IGC

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

The Resource Journal For IBM PC Programmers
PROGRAMMER'S Journal
Volume 6.1
January/February 1988

Contents

4 Note from the Editors

6 Letters

12 Consulting: Consulting Is an Attitude *by Paul Barkley*
From inside the walls of the workplace

16 On Graphics: Write Mode 3 of the VGA *by Michael Abrash*
Strange indeed—what's it for?

28 EMS: The LIM 4.0 Expanded Memory Specification *by John J. Newlin*
LIM and AQA get behind the new specification

38 Pascal: Routines to Access the Environment *by Tom Swan*
Tom takes us further into Turbo Pascal 4.0—environment variables

46 C: Time Is on My Side . . . Yes It Is *by Frank D. Greco*
Time your programs and function calls

52 FORTRAN: Porting Mainframe Software to the PC *by M. Steven Baker*
How to get ten pounds of stuff into a five-pound bag

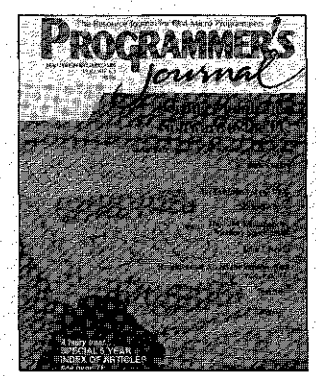
The Software Approach—FORTRAN Compilers Under DOS
A Hardware Alternative—the Definicon 32-bit Coprocessing Engines

71 A Five-year Index of *PJ* Articles
A special treat for collectors

76 Miscellaneous *by M. Steven Baker*
80386 development tools, 386 compilers, Turbo C V1.5, the FORTRAN 8X standard, Tom Swan's latest book

82 PJ Billboard

84 Advertiser Index



Five years old
and growing strong!
Artist Ralph Bentley
provided a sweet treat
for our fifth birthday.
The complete index on page 71
is just icing on the cake.
All this, and FORTRAN too!
Enjoy.

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

More Magic from Blaise.

Turbo C TOOLS™

Magic is easy with Turbo C TOOLS in your bag of tricks. New Turbo C TOOLS™ from Blaise Computing is a library of compiled C functions that allows you full control over the computer, the video environment, and the file system, and gives you the jump on building programs with Borland's new C compiler. Now you can concentrate on the creative parts of your programs.

The library comes with well-documented source code so that you can study, emulate, or adapt it to your specific needs. Blaise Computing's attention to detail, like the use of function prototyping, cleanly organized header files, and a comprehensive, fully-indexed manual, makes Turbo C TOOLS the choice for experienced software developers as well as newcomers to C.

Turbo C TOOLS provides the sophisticated, bullet-proof capabilities needed in today's programming environment, including removable windows, "side-kickable" applications, and general interrupt service routines written in C.

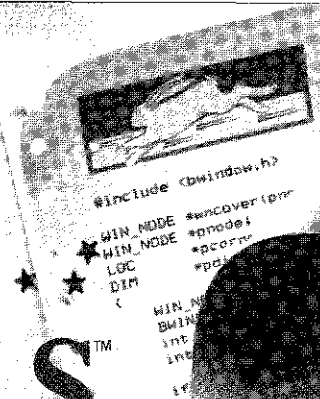
The functions contained in Turbo C TOOLS are carefully crafted to supplement Turbo C, exploiting its strengths without duplicating its library functions. As a result you'll get functions written predominantly in C, that isolate hardware independence, and are small and easy to use.

Turbo C TOOLS embodies the full spectrum of general purpose utility functions that are critical to today's applications. Some of the features in Turbo C TOOLS are:

- **WINDOWS** that are stackable and removable, that have optional borders and a cursor memory, and that can accept user input.
- **INTERRUPT SERVICE ROUTINE** support for truly flexible, robust and polite applications. We show you how to capture DOS critical errors and keystrokes.
- **INTERVENTION CODE** lets you develop memory resident applications that can take full advantage of DOS capabilities. With simple function calls, you can schedule a Turbo C function to execute either when a "hot key" is pressed or at a specified time.
- **RESIDENT SOFTWARE SUPPORT** lets you create, detect, and remove resident utilities that you write with Turbo C TOOLS.
- **FAST DIRECT VIDEO ACCESS** for efficiency, and support for all monitors including EGA 43-line mode.
- **DIRECTORY AND FILE HANDLING** support let you take advantage of the DOS file structure, including volume labels and directory structure.

In addition to Turbo C TOOLS, Blaise Computing Inc. has a full line of support products for Microsoft, Lattice and Datalight C, Microsoft Pascal and Turbo Pascal. Call today for details, and make magic!

Turbo C TOOLS supports the Borland Turbo C compiler, requires DOS 2.00 or later and is just $129.00



MORE BLAISE N U

Turbo POWER TOOLS PLUS *$99.95*
Screen and window management including EGA support, DOS memory control, ISRs, scheduled intervention code, and much more. For Turbo Pascal.

Turbo POWER SCREEN
COMING SOON! General screen management; paint screens; block mode data entry or field-by-field control with instant screen access. For Turbo Pascal.

Turbo ASYNCH PLUS *$99.95*
Interrupt driven support for the COM ports. I/O buffers up to 64K, XON/XOFF, up to 9600 baud, modem and XMODEM control. For Turbo Pascal.

PASCAL TOOLS/TOOLS 2 *$175.00*
Expanded string and screen handling; graphics routines; memory management; general program control; DOS file support and more. For MS Pascal.

C TOOLS PLUS *$175.00*
Windows; ISRs; screen handling; multiple monitors; EGA 43-line text mode; direct screen access; DOS file handling and more. For MS and Lattice C version 3.00 and later.

LIGHT TOOLS *$99.95*
Windows; ISRs; EGA 43-line text mode; direct screen access; DOS file handling and more. For the Datalight C compiler.

ASYNCH MANAGER *$175.00*
Full featured interrupt driven support for the COM ports. I/O buffers up to 64K; XON/XOFF; up to 9600 baud; modem control and XMODEM. For C or MS Pascal.

VIEW MANAGER *$275.00*
General screen control; paint screens; block mode data entry or field-by-field control with instant screen access. For C or MS Pascal.

RUNOFF *$49.95*
Text formatter for all programmers; flexible printer control; user-defined variables; index generation; general macro facility. Created in Turbo Pascal.

EXEC *$95.00*
NEW VERSION! Program chaining executive. Chain one program from another in different languages; specify communication areas; less than 2K of overhead.

TO ORDER CALL TOLL FREE
800-333-8067
TELEX NUMBER: 910339

YES! I want to make magic!
Enclosed is $_____ for _____ copies of _____.
☐ Please send me more information on your products.
CA residents add Sales Tax. Domestic orders add $4.00 for UPS shipping, $10.00 for Federal Express standard air.
Name: _____ Phone: (_____) _____
Address: _____ State: _____ Zip: _____
City: _____ Exp. Date: _____
VISA or MC#: _____

Turbo C is a trademark of Borland International

BLAISE COMPUTING INC.
2560 Ninth Street, Suite 316 Berkeley, CA 94710 (415) 540-5441

Number 2 on the Reader Service Card

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

The Resource Journal For IBM PC Programmers
PROGRAMMER'S Journal

Editors
Charles Aylworth, Erin O'Connor
Technical Editor
M. Steven Baker
Contributing Editors
Frank D. Greco, Tom Swan

Contributing Writers
Michael Abrash
M. Steven Baker
Paul Barkley
Frank D. Greco
John J. Newlin
Tom Swan

Publisher
Elizabeth Oakley
Production
Ralph Bentley
Illustration
Jim Carpenter
Advertising
Robert Tappan
Subscriptions
Gary Hoyer
Accounting
Bobbi Sinyard
Founder
Greg Estes

Business and Editorial Offices
Programmer's Journal
150 North 4th Street
Springfield, OR 97477
(503) 747-0800

Subscriptions: 1 year (6 issues) $24 ($30 in Canada and $40 elsewhere)
Change of Address
Programmer's Journal
P.O. Box 30160
Eugene, OR 97403

Articles and letters in Programmer's Journal represent the views of their authors and not necessarily those of the publisher. Letters to the editor become the property of Programmer's Journal and may be edited and published.

Code and listings in Programmer's Journal appear "as is," and Programmer's Journal makes no guarantee or warranty of their quality, performance, accuracy, or suitability for any particular purpose. Programmer's Journal assumes no responsibility or liability to any person or entity with respect to any damage caused or alleged to be caused by its use.

Advertising: For ad rates, deadlines, and placement, call Robert Tappan, Ralph Bentley, or Gary Hoyer (503/747-0800), or write to Programmer's Journal, P.O. Box 30160, Eugene, OR 97403.

Programmer's Journal (ISSN 0747-586) is published bi-monthly by Oakley Publishing Company. Second-class postage paid at Eugene, OR, and additional mailing offices.

POSTMASTER: Send address changes to Programmer's Journal, P.O. Box 30160, Eugene, OR 97403.

© 1988 *OAKLEY PUBLISHING COMPANY*

Note from the Editors

First, we need to clean up a few misunderstandings from last issue. I got the information about the Opus Hub a little bit wrong, and anyway, it's all changed by now. The Sysop of Blazer has decided to act as the primary hub for the *PJ* echo area. To establish a linkage, your Opus/FIDO sysop should call the Blazer directly, (503) 343-9724. Within the net, they are Net 152, Node 9 (152/9).

The Blazer has been set up to give new callers download access, so if you just want to leave us a note, or get the code files, you can call directly. Let us know how it's working out for you, and share any ideas you might have for alternate ways we can get you the code you need and want.

COMDEX is long gone by the time you read this, but it's still buzzing in my head. I won't tell you about it as the more news oriented magazines will have done the topic to death. All I will comment is "Wow, now I know what BIG means." It was well worth the trip. We met a bunch of potential authors who should be providing first class code in articles to follow. It was a lot of fun, but totally boggling.

I had a religious conversion at COMDEX. After a private audience with St. Silicon, I have seen the error of my ways. Now as a faithful member of the C.H.I.P. Church, I greet everyone I see with "Brother, is your data SAVED?" Who knows what will come of this. You should be hearing more of St. Silicon in months to follow. He's running for President, and is the first candidate I've heard who makes sense.—*Chuck*

COMDEX has come and gone. This time I didn't get fogged out (at least not 'til the way back), so I'll pass on some brief observations. Aside from the sheer size of COMDEX (90,000++ attendees), the most impressive aspect was the dominance of marketing in the microcomputer world. Lots of booths had to cost more than a quarter-million dollars for structure, staffing, and materials. Many were staffed by high-tech PR companies and musical entertainers as well as company staff. Real World's finely crafted $50,000 wooden multi-story booth certainly demonstated that COBOL is not a dead language on the PC (note their accounting packages).

Almost everyone was using 386 machines from Compaq, Multitech, and others to demo software. Chips and Technologies showed their 20 MHz chip set for 80386 motherboards that costs $100 in quantity. This is only $10 more than their high speed 286 chip set. Sure, there were PS/2s as well. The picture on OS/2 and its various flavors gets only cloudier for developers. IBM gave a large seminar and provided lists of announced developers for OS/2 software and PS/2 hardware. The OS/2 list included 66 firms of which only 21 had booths at COMDEX. That's a poor response rate from all those Microsoft OS/2 workshops attendees. OS/2 software was definitely not in the spotlight—one had to ferret it out. The IBM hardware list was much larger—about 150 firms. As I noted earlier, marketing and PR are driving the PC business. Whether OS/2 is the operating system for the future or the past, time will tell. To judge by COMDEX, the 80286 is definitely past. I don't envy small developers these days.—*Steve*

From *PJ* at the beginning of our sixth year—six good articles from as many good writers, code of course. We'd love to offer a baker's dozen, and if we last twelve years, I'm certain we will. Yes, you could ask for more, and it's hard to publish something for every last one of you in one issue, but this one is up to date and balanced. From our Reader's Survey, I've learned that some of you are delighted to subscribe to a magazine that publishes even one pertinent article per issue. There must be a paucity of good writing out there, but there ain't no paucity in here. A lot of you as a matter of fact have a good chance of finding one or two or even three articles of interest to you in this one.

I feel good about this issue. It's up to snuff. The Letters column's got some heat, and Steve Baker's FORTRAN opus spills out into a Miscellaneous column that even so lives up to its name. Our fifth-birthday index of *PJ* articles (go to the end of the book) should step up back issue sales so that we can bring you even more good writing.

The boys went down to Vegas and ran into at least a dozen potential new writers for *PJ*. We're excited about their levels of expertise and the range of topics. Look forward to more good issues of *PJ*.—*Erin* ♦

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

DEBUGGING SWAT TEAM

Order Eco-C88 Rel. 4.0 New Modeling Compiler and get C-more at no extra charge!

Seek and Correct

You already know that fast compilation does not mean fast program development. Backing up for bogus error messages and removing the bugs takes time. Eco-C88's "Seek and Correct" three - way error checking finds even the most elusive bugs, clearing the path for swift program development.

Double Barrel Error Checking

Eco-C88 nails syntax errors cold and tells you about the error in plain English. And there's no avalanche of false error messages, either. Other compilers can generate up to four times the number of error messages actually present; they leave it up to you to guess which ones are real. You'll be more productive with Eco-C88 because there is no guess work.

Eco-C88 provides ten levels of semantic error checking. You can select from almost no checking to the fussiest you've ever seen. Eco-C88's "picky flag" finds subtle errors that slip by other compilers.

Eco-C88 also features:

- All data types, plus ANSI Enhancements
- Robust library, including many new ANSI functions
- CED editor with online function help, split windows, compile-edit-link capability
- New, expanded manual with sample programs for the library functions

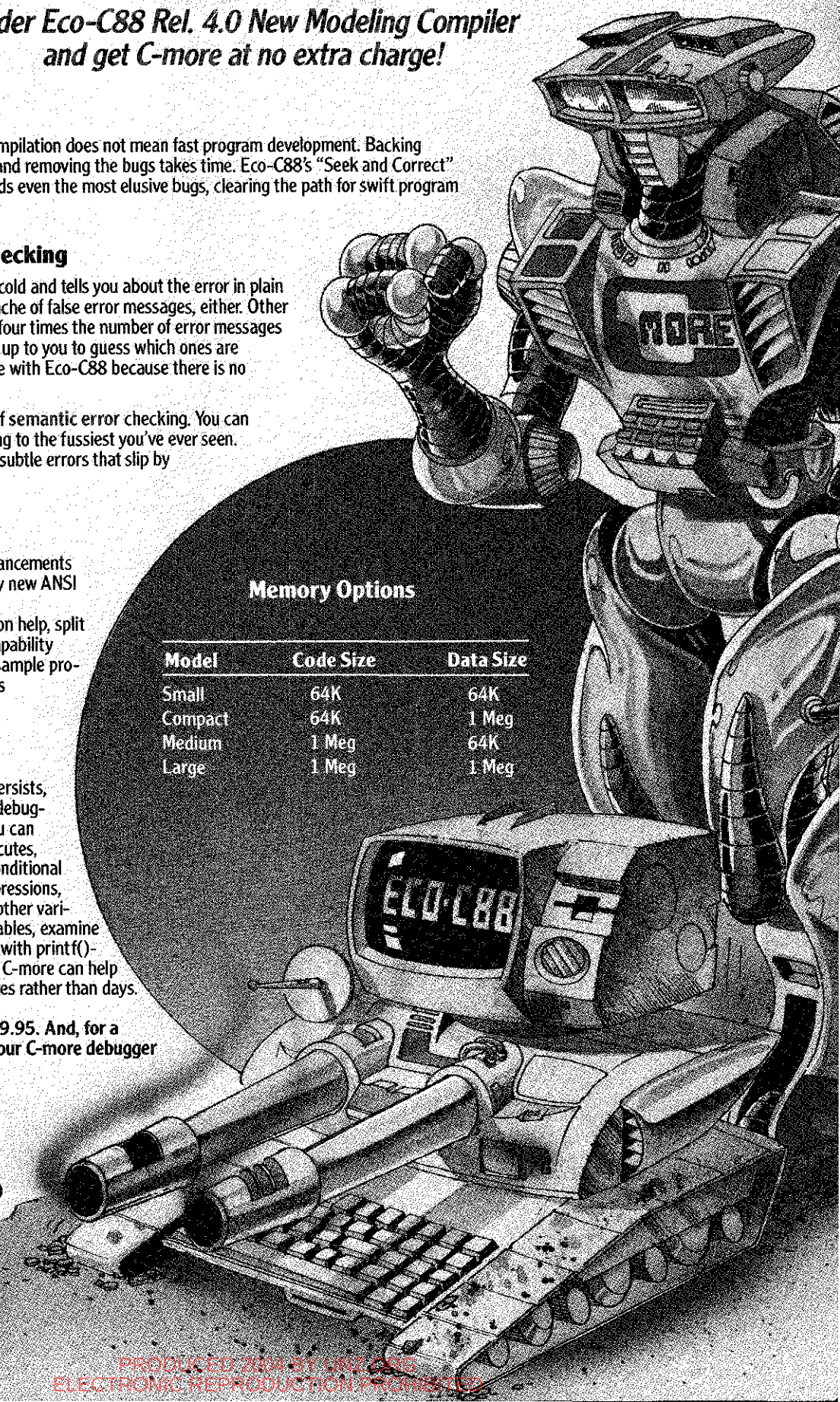
C-more Source Code Debugger

Finally, if a really nasty bug persists, put C-more, our source code debugger, to work. With C-more you can watch your program as it executes, single-step it, set simple or conditional breakpoints, test complex expressions, use variables as indexes into other variables, initialize and trace variables, examine CPU registers, display results with printf()-type options and much more. C-more can help you track down bugs in minutes rather than days.

The price for Eco-C88 is $99.95. And, for a limited time, we'll give you our C-more debugger at no extra charge.

Memory Options

Model	Code Size	Data Size
Small	64K	64K
Compact	64K	1 Meg
Medium	1 Meg	64K
Large	1 Meg	1 Meg



Ecosoft Inc.
6413 N. College Ave.
Indianapolis, IN 46220

(317) 255-6476 (Tech Info)
(800) 952-0472 (Orders)

Number 8 on the Reader Service Card

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

More Properly . . .

Overall, Tom Swan's TSR ("The Case of the Invisible CapsLock," *PJ*, Volume 5.5, September/October 1987) does not serve as a reasonable example of how to write a TSR though it does do something useful. To me, Swan appears to be less skilled in writing MASM than Turbo Pascal.

1. The program uses interrupt 27h to terminate and stay resident, contrary to the Microsoft recommendation to use interrupt 21h function 31h for this purpose. This will allow CapsLock to run under DOS 1.x, however.

2. As a matter of coding convention, the int9 procedure should be defined as "PROC far." Then it would not be necessary to code the far jump at the end in hex. The jmp op code would suffice. As written, the far jump at the end of the int9 procedure is modified code, a dirty programming practice to be avoided. It's far better to place the double word defined at "address:" some distance from the executable code, then do "jmp dword ptr address."

3. For a system with a VGA, MCGA, or EGA video controller, the cursor may be out of proportion in video modes 2, 3, and 7 since the newer controllers map the older CGA or monochrome text into 8 x 14, 9 x 14, 8 x 16, or 9 x 16 pixels, depending on the mode and controller. I know of no way to remedy this in CapsLock without introducing the overhead of a video BIOS call, interrupt 10h, function 0Fh, to determine the display mode.

4. But worst of all, CapsLock assumes that it will always be resident—forever. As a general rule, I avoid using TSRs that cannot remove themselves from memory, or be removed from memory. I also do not write TSRs that cannot remove themselves from memory. Given the current constraint that DOS has only 640 Kb of memory available to all programs, it is appropriate that TSRs be written to be removed, to eliminate ram cram without forcing the PC user to reboot. Yes, there are now programs that manage ram cram, but they too take up more memory, thereby worsening the situation. Unfortunately, there are all too many TSRs on the market, for sale, not free as CapsLock is, that do not extricate themselves from memory. Among those I have come across recently are the TSRs supplied with the Logitech mouse, the Quaid Analyzer, and a simple measuring package called StopWatch.

To follow up on another subject in the same issue—John Newlin's Turbo Pascal TSRs that use expanded memory—Version 4.0 of the LIM Expanded Memory Specification is now available to interested developers for the cost of a phone call to Intel Customer Support at (800) 538-3373. It is document number (Intel's?) 300275-004 and dated August 1987. The LIM 4.0 spec contains clear examples in MASM, Turbo Pascal, and Microsoft C. Its 204 pages also explain EMS programming musts and guidelines, document every EMS call, give function/status code cross references, and contain both glossary and index. Thoroughly professional. Three cheers for Intel. They understand the need to provide the PC software development community with the information needed to write additional EMS programs, indirectly helping sales of the Above Board.

Ben Myers
Harvard, MA

See John Newlin's article on the new LIM spec in this issue.—Ed.

;,(* *), or {}, Please

My compliments on an excellent issue! I must admit to a bias, however, as this is the first *PJ* I've ever read. I want to pass on a few comments that may help out in future issues.

In the listing for John Newlin's "TSRs Using Expanded Memory with Turbo

Form-Fill Toolkit

Product: C Library
Features:

Price: $59.95

Cursor Control
Sound & Printer Control
Fast Screen Access
Keyboard Access
Edited Field Entry
Pop-Up Window Support
Display Text Windows
Menu & List Selections
Form-fill Routines

Compiler Support

Microsoft 4.0, : Y
Microsoft 5.0 : Y
Quick C : Y
Turbo C : Y
Other:

↑↓ Scroll F1 Help F2 Save F4 Cancel Esc No Save

Provide a consistent user interface for your programs with over 100 documented functions.
SPECIAL! Includes powerful multi-record data collection module with fully commented pristine source code.

Anji Software, Inc.
P.O. Box 297
Cedar Grove, NJ 07009
(201) 239-4328

- Library Source Code - $119.95
- Specify compiler when ordering
- No Royalties required
- Demo Diskette $10.00

We accept Visa and M/C
NJ Residents add Sales Tax
Shipping $3.00 Domestic
$10.00 Foreign

Number 162 on the Reader Service Card

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

Pascal" (Volume 5.5), I occasionally had trouble distinguishing between comments and actual code. Mr. Newlin added comments to some areas of the code that were not actually part of the code and were not delineated as comments by ;, (**), or { }. For most people this probably wasn't a problem, but for the assembly novice (I count myself as one) it can lead to confusion. Please don't misunderstand—it was kind of Mr. Newlin to comment the program so thoroughly for those of us who don't program regularly in assembly—it was just the non-delineated comments that threw me off.

In the same article, a gray box showed the LIM EMS functions and gave a number to call for full documentation. Intel informed me that there is a whole new system (4.0) for these calls and promised to send out a complete description. For anyone who hasn't called Intel and is still interested, typing 474 when prompted for questions and answers by their phone system's main menu will send you immediately to the proper message.

W. Brett Fishburne
College Park, MD

OS/3?

The most recent issue (Volume 5.5) was interesting, but (a little carping criticism here) I didn't find Christian Walker's closing comments on OS/2 to be very well informed. It has its weak points, but he didn't identify them! Also, much of the point of doing a conversion to OS/2 now is that the subsequent conversion to OS/3 (or whatever they call the 386 32-bit version of the system) should be extremely easy, due to the call based API. Well, he obviously has an axe to grind (as do I), but I think if he really spent some time writing programs for OS/2 he'd probably be very impressed with it. I haven't talked to any programmers who are working with OS/2 who didn't like it (a self-fulfilling prophesy possibly). On the other hand, the Presentation Manager looks needlessly complex to me.

Ray Duncan
Marina del Rey, CA

Christian Walker Replies

Obviously Mr. Duncan is an advocate of OS/2, or he wouldn't have written a book about it. It is inevitable that any software—especially from IBM and particularly when it endeavors to communicate with IBM mainframes and attempts to solve the desktop accessory problem—will appeal to some particular subset of the general computing population. My concern is that the product as it has been announced cannot possibly meet the needs of a particularly large subset of that population, beset as it is by the specific weak points mentioned in the article (size, expense, 32-megabyte filesize max and 64K byte segment max). I'm sure that, as Mr. Duncan implies, there are additional weaknesses, but the above mentioned ones alone significantly reduce the usefulness of the product.

As for Mr. Duncan's optimistic belief that software for the OS/2 will readily convert to OS/3, I can only point out the obvious fact that no call interface, no matter how standard, can hide the fact that pointers must have a different size in a native 386 operating system. Any code that is pointer-size dependent will clearly require significant alterations to shift to the new machine. But it is hard to argue about OS/3 since IBM hasn't even announced when they are going to announce it. In other words, it hasn't even reached the status of vaporware yet.

Christian Walker
Hanover, NH

Park Your AT into Oblivion

Let's say you want to park the heads on the hard disk in your AT prior to moving it. You find an available Diagnostics disk (this one for the XT) and use it. The next time you want to boot the computer, you get nasty messages about "unrecognized drive," etc. In short, your hard disk is gone. No amount of using DEBUG, Norton, or other utilities will get it back.

The problem is that the older versions of SHIPDISK.COM on the XT Diagnostics disk actually contain instructions to format the current track in the event of an unexpected error from the hard disk. The villains are dated 11-02-82 and 02-26-83 and are 421 and 436 bytes, respectively. To see this nefarious code, use DEBUG as follows: DEBUG A:SHIPDISK.COM to load the program into memory, U190 to unassemble the active portion of the program, and then look for the instruction "MOV AX,0506" (it should be around location 1F0). The "0506" specifies a format operation with an interleave factor of 6.

This is correct (though questionable) logic for the XT fixed disk controller, but the AT does it a different way. The "05" command for formatting a track is the same, but the interleave is defined in a table pointed at by the ES:BX registers. Unfortunately, the XT version of SHIPDISK makes no attempt to set these registers, so they are pointing somewhere in memory (probably to the

VERSION 3.0
RASP®
GRAPHIC ANIMATION SYSTEM FOR PROFESSIONALS
GRASP is a simple graphical programming language used to create and run animated graphics, demonstrations, tutorials and presentations, or add visual effects to other programs. GRASP allows images to be captured from other applications, or be created using Pictor, a built-in paint program.
$99.00
■ Single command controls animation sequences. ■ 16 picture buffers. ■ 128 clipping buffers. ■ 64 commands. ■ 25 different fades with limitless combinations. ■ Simple ASCII file format. ■ Execute custom programs from GRASP. ■ Call GRASP functions from your applications. ■ Supports VGA, CGA, EGA, HERCULES, HERCULES InColor, text modes.
Paul Mace
SOFTWARE
800 523-0258
503 488-0224
Number 80 on the Reader Service Card

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

Program Prefix). Naturally, what the AT controller finds there is "garbage," and the result is that Track 0 on the hard disk is formatted with unexpected sector numbers.

During the booting process, ROM BIOS wants to find a Master Boot sector on track 0 at sector 1. No sector 1, no Boot—therefore, no acceptable device. The hard disk is "out to lunch."

Presently, no commercially available software can fix this problem. What's needed is a "low level" format of just track 0 followed by a successful FDISK operation to rebuild the Master Boot sector and redefine the Partition Table—no simple matter since you have to search for the starting track of each DOS partition in order to correctly rebuild the pointers. With luck, everything will be contiguous and FDISK will match the original. The usual answer is to abandon all the programs and data on the disk, perform a "low level" format, and start over with an empty disk.

Prevention? If you are in a mixed XT and AT environment, my advice is to collect all the XT Diagnostic disks, lock them up, and carefully control how and when they are used. But a common technique is to make a "PARK" directory on the hard disk, place in it the COMMAND.COM and SHIPDISK.COM programs, and then build a batch file to invoke COMMAND when you want to park the heads on the hard disk. It might be a good idea to inventory all your ATs and make sure that no XT versions are there. It would be a hard to detect form of sabotage for someone to put the dangerous XT version on an AT. Once the "park" routine is performed, all access to the hard disk is gone, hiding the cause of the problem.

Grey Staples, CDP, CCP
Information Repair Services
Scottsdale, AZ

"Developing my application in C would have taken 6 months to a year, but in Actor it took 2 months." —*Brian Fenske, Boeing Commercial Airplane Company*

"To C or not to C..." ACTOR

Actually, you don't have to make the choice. Once C was ideal for all PC programming. But it has been complicated by windowing and graphical interfaces. Now windows development with C is difficult, time-consuming and error-prone. You need a new language that simplifies windows programming. Introducing Actor®.

Actor is the first interactive object-oriented language made for commercial development. Its powerful browsers, inspectors and debuggers give you more insight into a windowing environment than C ever will. But your C work is not lost. C libraries can be linked to Actor. Plus, its procedural syntax is easy for C programmers to learn.

Actor comes with windowing classes built in. Customize Actor's classes to create standalone windowing applications. And objects give you another layer of independence for a smooth transition to OS/2 and Presentation Manager. It's the quickest and easiest way to write a windowing program.

"You can write Windows programs much faster with Actor than with C or assembly language."
—PC Magazine, June 9, 1987

Tech Specs

NOW Release Version 2.1

- Runs with Microsoft Windows 1.04, 2.0 and 386. Extended memory under 2.0 and 386.
- Pure, single-inheritance object-oriented language, incrementally compiled.
- Dynamic linking to C, Pascal, Assembler, or Fortran libraries. Pass data in C structures.
- Pascal and C-like syntax.
- Programming tools: Browser, Inspector, Debugger, File Editor.
- Full access to MS-Windows systems calls, multitasking, and DDE.
- Fast device-independent graphics: lines, shapes, icons, cursors, bitmaps, metafiles, Turtle graphics, sample control language using YACC.
- 150 classes, 1500 functions, fully extensible.
- Window styles: tiled, overlapping, popup, child, edit, dialogs. Controls: list boxes, scroll bars, buttons, check boxes.
- Data structures: stacks, arrays, queues, lists, dictionaries, sets, sorting, hashing, intervals.
- AI support: frames, symbols, dictionaries, lists, symbolic programming, functional arguments. Parsing and lexical analysis YACC compatible.
- String manipulation: substring, concat, append, insert, remove, search.
- 643-page manual includes tutorial and reference.
- No license fees. Generates stand-alone applications.
- Fastest interactive OOL available.
- Fast incremental garbage collector.

Actor $495 • Academic price $99 • Academic site license $99 • Manuals for site license $35 • New! **Language Extension $99 • Shipping $5 US, $25 Int'l**

The Whitewater Group
Technology Innovation Center
906 University Place, Evanston, Illinois 60201
(312) 491-2370

Actor is a registered trademark of The Whitewater Group, Inc.

Number 73 on the Reader Service Card

Read *PJ* for the Ads?

I hardly ever get around to writing to magazine editors, but I thought you'd like to hear some good things about a couple of your advertisers. I do a lot of Turbo Pascal programming on a PC, so I bought a copy of TMARK from Tangent Designs. This little program lets Turbo continue compilation after an error has been found and fixed—right where it left off. Wonderful! And I just received a copy of the Turbo Pascal source code from CC Software (my curiosity got the better of me). I'm sure it's only a reconstruction (not what Borland works from), but it truly is amazing. Now I have to find time to learn "all about" how compilers are written. Both programs were delivered promptly and at reasonable prices. Let's have more of them!

Lest you think I don't read your magazine—I devour every Turbo Pascal article you print. I find many helpful hints (some I have already learned the hard way). I know C is the up-and-coming language, but I just can't read the stuff.

Michelle Adolph
Concord, CA

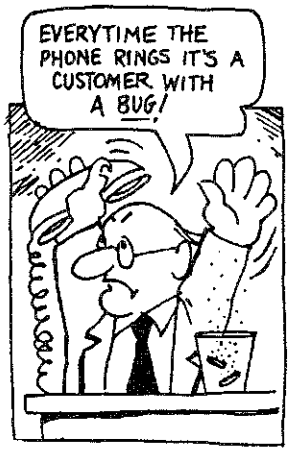
Macrobnk

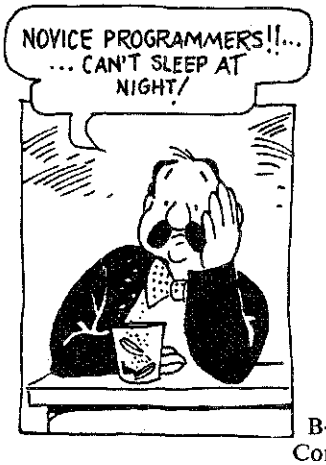
We are avid users of macros to aid our word processing tasks. We define macros as prerecorded keystrokes that are fed into programs one at a time when a signal is given. Although books on the use of macros in spreadsheets are available, less attention has been given to their best use in word processing. Accordingly, we have started a Macrobnk, an exchange service for word processing macros so that good ideas can be disseminated to others. If you send us an MS-DOS disk (5.25-inch or 3.5-inch, any density), containing macros you use, we will incorporate your macros into the collection we

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

Got Software Problems?

Get TURBO GhostWriter





Turbo GhostWriter is an application generator that creates 80% of your custom application automatically -- as a Turbo Pascal Program. **That 80%** is done swiftly in just 10% of the time it takes to code it yourself. **That 80%** contains all of the "hooks" to add custom features such as table verification, security, currency conversion, importing files from other languages, etc...

Now it's up to you to write the code to relate one file to another. The relational model shows you how and it only takes a few minutes. End to end, all programming for 5 relational files should take less than 30 minutes. It's far superior to doing the whole thing from scratch. Your user sees a consistent interface, all programs function the same way, and your code is 100% error-free Turbo Pascal.

ASCII wants to give you a chance to try Turbo GhostWriter. For a limited time you can buy a fully functional version for just $99. This version will create flat files such as mailing lists, telephone files, inventory files, simple payroll files, etc. The utilities are the same as for the full-blown product, but are modestly packaged, and support is limited. It's the perfect way to see if an application generator is for you without risking a cent.

Now $99.00

	Full Version **$289**	Starter Version **$99**	Cost of Upgrade for $99 version
B-Tree File Manager	Yes	Yes	N/A
Context-Sensitive Help	Yes		50
Command Window	Yes		50
Relational Model	Yes		150
Free Support	No Limit	1 Hour	100
Manual	Cloth D-ring	Paperback	50
All Upgrades	...	...	225

Orders & Information 800 227-7681

30 day money-back guarantee*(less $14 s/h)*

YES! Send me___copies of Turbo GhostWriter today.
Name _____
Address _____
City,State,Zip _____
Phone _____
MC, Visa, or Choice Card
#(or write COD) _____
Expiration Date _____
Circle version $99 $289
ASCII - 3239 Mill Run - Raleigh, NC 27612
800 227-7681

Number 170 on the Reader Service Card

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

have already received and send you a copy of the complete set. Don't feel that your macros have to be especially complex. We are interested in all the macros you use, particularly in the simple ones you use everyday. Please consider anything you send us to be in the public domain as we will make it available to other users of macros without charge.

Rollie Cole and Paul Sommers
Macrobnk
14022 23rd Avenue, NE
Seattle, WA

Turbo Pascal 4.0 Preview Update
There were a few errors in my article, "Turbo Pascal's Renaissance" (*PJ*, Volume 5.6). In the Numeric Data Type table, the first two entries should read

Data type	Bytes	Range
Byte	1	0..255
ShortInt	1	–128..127

I also mistakenly implied that the compiler emulated IEEE real number data types in the absence of a math coprocessor. The Single, Double, Extended, and Comp real-number types require a coprocessor chip. Without the chip, you can use only the 6-byte Real type.

The LST and CON file devices do not exist. You can open text files PRN for printing and CON for the console, but there are no predeclared file variables as there are in 3.0.

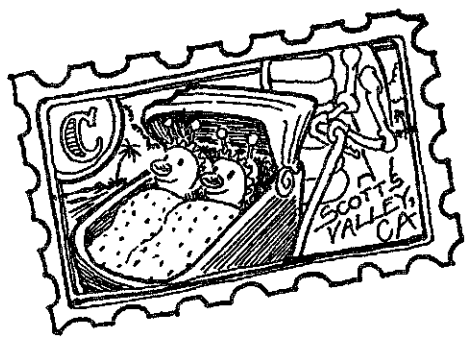
Tom Swan
Lititz, PA

Borland and the ANSI C Standard
Rex Jaeschke reported a pair of related bugs in Turbo C 1.0 in which the compiler does not properly handle the predefined ANSI macros (*PJ*, Volume 5.6). These bugs have been reported and fixed in the new Turbo C 1.5. All of the errors he described have been fixed in this release.

Borland International has made a concerted effort to support the ANSI standard as it has emerged. We have, however, tried to balance the introduction of new features with the need to give users a consistent product from release to release. As a result, we have deliberately chosen to avoid implementing sections of the standard—most of which are in the runtime library—that are likely to change before the final adoption of the standard. We are committed to supplying a conforming compiler as soon as that is possible. Since the ANSI committee is still not done with their work, we must wait to finish the job of making Turbo C conform to the ANSI standard.

Some of the issues concerning symbol names in the library will be dealt with before the final acceptance of the standard since the set of reserved names is not likely to change substantially. This area is complicated by the need to maintain compatibility with other existing C libraries that are not ANSI compatible. This particularly applies to the function **matherr**, which has a long history of use, but has been left out of the ANSI standard.

Brad Silverberg
Vice President, Engineering
Borland International
Scotts Valley, CA ◆



If you can find better sort/merge software, buy it!

Sortex is a flexible, general purpose sort/merge facility which provides you with the ultimate in performance and reliability

- You can run Sortex from the DOS prompt, batch file, or applications written in Assembler, BASIC, COBOL, FORTRAN, Pascal, or C.
- You can tell Sortex what to do by using the menu/panel facility, control file, or command line.
- Sortex can process most of the commonly used file types — both fixed and variable length records.
- Sortex supports a wide variety of data types.
- Sortex imposes no practical limitations. Files with two billion characters, records & fields with thirty two thousand characters each, and unlimited number of keys can be handled by Sortex.
- Sortex allows records to be chosen selectively for processing.
- Sortex provides comprehensive error reporting, so you can quickly locate and eliminate the source of errors.
- You can direct Sortex to log all the program messages and to issue progress messages while processing your data.

No Royalties Not Copy Protected 30-Day Money-Back Guarantee

Only $99.95 Introductory Price — Limited Time Offer

Systemat CORPORATION

1915 W. Orangewood Avenue
Orange, California 92668
Telephone: 714 978 0480 Telex: 910 250 0674

Number 171 on the Reader Service Card

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

Attention Programmers!

HERE'S THE QUICK, EASY WAY TO MAKE SCREENS, DEMOS AND TERRIFIC PROGRAMS!

DisplayExpress™: A fast, efficient and easy-to-use screen manager.

DisplayExpress™ helps you make efficient screens as easily as you would type a letter. Such screens take little space (less than 2 kbytes in the average) and are displayed instantaneously.

Colors...Character graphics...Editing power...Automatic drawing commands...And a very compact run-time support (source included) - - 6 kbytes only, yet it has all you need to load and display screens and menus!

The package also includes the DemoExpress® Viewer which allows you to make demodiskettes and slides quickly and easily. Ideal for your presentations, tutorials and continuously running demos.

C-Display Librarian™: The professional programmers' choice.

C-Display Librarian™ provides complete support for the man-machine interface. Perfoms BIOS video, screen editing, data entry and validation, keyboard control and bit mapped graphics.

Modular design...Small size...Great speed... Professional documentation...Full source... Compatible with Microsoft C, Pascal and Fortran, Lattice C and Borland International Turbo C compilers.

"Tremendous productivity tools"
*(**PC Week** October 27, 1987)*

"Many applications will benefit from the quick development times and minimal programming required to implement a finished, quality product"
*(**IEEE Software** November, 1987)*

"If you want to include bit-mapped graphics in your application, then this might be the package for you"
*(**Computer Language** June, 1987)*

DisplayExpress™ $69

C-Display Librarian™ $69

C-Display Manager™ $118

A money-saving combination that includes, in a single package, both the DisplayExpress and the C-Display Librarian

$20 off

No royalties on programs or demos!
Free programming examples with source!
Free shipping within the United States on all prepaid orders!

Satisfaction guaranteed or your money back!

YES! **I want to make screens, demos and terrific programs right now! Please rush me:**

QTY.	PRODUCT NAME	UNIT PRICE	TOTAL
____	**DisplayExpress™**	$69	____
____	**C-Display Librarian™**	$69	____
____	**C-Display Manager™**	$118	____
	Sales Tax (New York Residents Only)	$	____
	All Foreign Orders: Please Add $10.00 per Product and Make Payment by Bank Draft Payable in U.S. Dollars and Drawn on a U.S. Bank.	$	____
	TOTAL	$	____

Diskette model (please check only one)

☐ 5¼ inch for IBM PC/XT/AT and truly compatibles.
☐ 3½ inch for IBM Personal System /2.

Mode of payment (please check only one)

☐ Payment Enclosed Herewith (**FREE** shipping within the United States on all prepaid orders)
☐ Please Ship C.O.D. (Shipping and C.O.D. charges will be added to the order)

NAME _____
TITLE _____ TEL.: _____
COMPANY _____
ADDRESS _____
CITY _____
STATE _____ ZIP _____ COUNTRY _____

SYDETECH
SYSTEM DEVELOPMENT TECHNOLOGIES INC.
43-23 COLDEN STREET, #17C, FLUSHING, N.Y. 11355

For COD orders or the name of the nearest dealer please call (718) 886-0221

Number **74** on the Reader Service Card

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

Consulting Is an Attitude

by
Paul Barkley

In the last few months I've had an opportunity to talk to quite a few people in the data processing field who are not consultants. After hanging around with nothing but consultants for a number of years, I'm starting to talk to a lot more "normal" people. What I'm hearing is hair raising. There are a lot of unhappy people in the regular work world.

The Workplace

An analyst, the systems kind and not the head shrinking variety, came into my office and said that the night before she had had a wonderful dream. In her dream she realized that it was Friday, and when she woke up she felt terrific. Then she remembered that in real life it was only Monday morning. "Shoot," is more or less what she said, prying open her eyelids and facing another week of very unsatisfying labor. You don't need Sigmund to tell you that wishing Monday were Friday is an unhealthy attitude toward work.

She was telling me this when she noticed on my desk a design document folder. With the instinct of a mother, she recognized that it contained a system she had designed. Flipping it open (hey, why don't you help yourself, my desk is your desk, make yourself at home), she flew into a rage because it was covered with red ink from one end to another.

"This is so denigrating!" she shouted, obviously upset at having her work edited. "Everyone just assumes that everything I do is wrong!"

What could I say? There certainly was a lot of red ink on her document, but an awful lot of it came from the same error made repeatedly. The editor who watches over the technical writers gets a crack at the design documents, too, after I check them to make sure the designs won't give either the programmers or the users heart failure. I also check to make sure that all the documents use the same screen standards and so forth.

The editor, on the other hand, checks to make sure the writing itself is clean. One big no-no is to say, "The screen shown in Figure 1 . . ." It should say, "The screen shown as Figure 1 . . ." (The argument, in case you're interested in this earth-shaking issue, is that a piece of a screen can be contained in a figure, but the whole screen is the figure and can't be in it. Big deal.)

"I've written many papers for scholarly journals, as well as dissertations, and that's absolutely ridiculous. We always say 'in' when we refer to figures," she continued.

It's days like this that make you wonder how anything gets done in a group of people greater than about two.

"Listen," I said, "what's the big deal? I don't make the editorial policy around here. That's why we've got an editor. If she says it's supposed to be done that way, I bet she's got a good reason. She can either show you the page reference in the *GPO Style Manual*, or she can quote you a sound reason for deviating from that standard. She'll be happy to talk it over with you—just ask her."

"Besides," I continued, "what do you care? It doesn't change the meaning of what you wrote, and you don't even have to make the changes. The secretary will do it for you—that's her job."

I begin referring to the standards as the "standards du jour," which makes me feel better.

"This is just so," she paused, "so demeaning!"

I think it's clear that someone who feels cheapened by having her work edited to conform with standards is someone who has her ego tied up in the wrong place.

But this isn't an isolated case. I see a lot of interpersonal problems. Of course it's possible that this particular company really is run by an assortment of cretins, idiots, perverts, and people with severe psychopathological problems, as many employees allege. I may have even, ahem, advanced that theory once or twice myself. If in fact this is such a terrible place, that would certainly explain a lot of the complaining and unhappiness. Of course that doesn't explain all the cheerful behavior that I see, too.

"What does all this have to do with consulting?" you ask, wondering when I'm finally going to get to the point.

Well, it seems to me that there's a big difference between these employees and the consultants I know. Sure, consultants complain about clients all the time. I've talked about the value of consultants getting together periodically just to let off steam. Obviously, consulting is not a problem-free profession.

Trapped

What seems to me to be the big difference, though, is that employees feel trapped and helpless. These people see their entire business careers molded by forces over which they have no control. Consultants generally feel much more in control because they're outsiders. There's an arm's length relation with clients that says, "Respect me, and respect my profession. I'm a highly paid consultant. Don't waste my time, and I won't waste your money."

Ever see too many animals crowded into a cage? It's not a pretty sight. Their behavior turns aggressive. In animals this translates to chunks of missing hide. In humans, if the crowding is minor, you see only surly behavior, snappishness, and general irritability. If it gets worse, for whatever reason, you may see handguns and total devastation.

Okay, so we're not talking about physically crowding employees. Or are we? What's the effect of putting four programmers in a single room? Do you think they feel crowded when one of them is playing a radio and one is trying to talk on the phone?

What can we do to make this situation even worse? How about insisting on some tight standards, and then changing them periodically? That should do the trick.

In our group we've had a problem with a couple issues, like the placement of the prompt on menus. Someone who has the authority to overturn a decision wasn't paying attention last July when I issued the standards, which of course included menus. He signed off on them.

Then, a couple weeks ago this person notices what the screens look like. "This is not right," he says, sketching the new menu design. Presto. A new menu standard is issued. (We won't even talk about the several changes in the screen numbering scheme—that subject is still too painful.)

Well, now, that's pretty annoying to me, but it's real annoying to the programmers who have to tear out the code. They take that sort of stuff personally. (Am I ringing any bells for you?) In fact, some of them are quite angry about it.

Well, that's life, I tell myself philosophically. I begin referring to the standards as the "standards du jour," which makes me feel better. In fact, I begin to see the humor in the

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

situation, and a few other people can laugh at it, too.

But not everyone is laughing.

The analyst who felt denigrated began shouting that every time they tell her to do things one way, they turn around and change their minds the next day. "Why don't they make up their minds?" she yells.

"Well, first of all, we're not talking about a nameless, faceless 'they.' We're talking about a person," I said, naming names. "Next, let's look at some concrete examples."

The analyst didn't want to look at concrete examples. In fact, the analyst didn't much want to get anything resolved. She was worked up to a real lather. I watched her mouth closely, hoping she wasn't going to suddenly bite me. You can get some nasty germs that way.

When I later investigated the situation, I found a couple examples of "in figure" the editor had missed. She wasn't being inconsistent—she just screwed up and missed a few. Writers really hate inconsistent editors, though. It's okay for writers to make a mistake—that's what editors are for.

There are a couple things to learn from all this, besides the need to get a very large bar on your office door to keep people with problems out of your face. First, employees generally have some real basis for their dissatisfaction. Life is not perfect. In fact, a lot of days are pretty nutso. So, there are always going to be things to complain about.

Second, each person has to decide if some nonsense at work really affects her or his own sense of self-worth. Was the editing of the analyst's work really a degrading experience, or did she just feel cheapened without much reason? In fact, was the editing really the source of all this intense dissatisfaction, or did her problem stem from something else and just manifest itself in this way? Is editing the real problem, or is it something else?

Third, are you getting enough out of your job to compensate for the irritating things that happen every day? Most people have at least a few days every now and then when clearly the answer is "hell no," and they begin to think seriously about other employment. When those days outnumber the pleasant days, then of course it is time to move on, and you should devote your energy to leaving, not complaining.

Fourth, have you really tried to do anything to change the situation? Of course some situations are truly hopeless. It's particularly difficult to change people's attitudes. If you work for a sexist pig, you've either got to decide that he's (I guess there could be female sexist pigs, but I've never met any) the one with the problem and you'll be able to tough it out, or you have to decide to leave. In some cases, though, you can help shape the workplace to remove many of these irritants, or you can take steps to stay away from the source of irritation.

At this particular site, it's mandatory to wear security badges. These badges are either attached to a necklace (made from the same lovely chain they use to keep ball point pens from disappearing at the bank), or they're attached to double-jointed clips. When you hang the badge around your neck, it bangs into everything when you lean over, so if you're "cool," you tuck it in your shirt pocket. You now have what looks like a very long, very ugly watch fob around your neck, disappearing into your pocket with what must be a very flat pocket watch. Or, you can use the clip, which causes your pocket to sag progressively outward until the badge flips over completely. You could be a total nerd and clip it on your belt, I guess.

Okay, so the badges are annoying. I forgot to mention that almost no one in the building works on anything classified, either, which makes it real silly. I say take those six people on the secret project and stick 'em in their own building somewhere else. Give them an office at some military site behind barbed wire, and let military police handle security.

We're talking about a pretty silly nuisance in the cosmic scheme of things.

Now here's the really annoying part. We have a security officer who roams around looking for people who don't have their badges displayed. "Out of your shirt pocket!" she barks. How would you like to have her job? Considering that she often says this when you're in the hall about six feet from your own office, surrounded by the people who work for you and who obviously know you aren't a KGB agent, you have to wonder about the importance of displaying your badge.

"Solutions"

I figure there are four things I can do. The first that comes to mind is that I can assault her. That seems like the perfectly logical response to this obnoxious behavior. She's smaller than I am, and she doesn't carry a handgun. Of course, I don't know how someone gets to be the security officer, and I don't want to find out the hard way about all her martial arts training. I also don't want to find out suddenly about her 12' 7" homicidal boyfriend with the razor collection. There are obviously a number of problems with this violent approach.

The second thing I can do is to wear my badge all the time like a good boy. This doesn't have much appeal either.

The third thing I can do is to get really mad about this. Wearing a badge is terribly degrading. It makes me feel like a piece of meat. It demeans my professional status. It's worse than punching a time clock. It wrecks my clothes. It's ruining my life. I can start having dreams about badges, and I can decide that any company that makes you wear security badges for no good reason must be a company that treats its people like dirt. I could do this, but I find it difficult to sustain the rage for more than three or four minutes.

The fourth thing I can do is to ignore the situation, to trivialize it, and to laugh at it. We're talking about a pretty silly nuisance in the cosmic scheme of things. It's about on a par with the hard rain that starts a few minutes before you want to leave the building.

Part of this strategy is also to avoid the irritation. I can't stay out of the hallways, but I can make it a point to display my badge properly when I have to wander into other parts of the building where I am in fact not known. I especially behave when I have to walk by the security office. It's like slowing down to 55 when you cruise by the state police barracks on the toll road. It just seems prudent.

The analyst could have done the same thing. She could have shrugged her shoulders and let it go. It didn't make any more work for her. She wasn't submitting the document to the Pulitzer Prize committee. In the overall scheme of things, it didn't really matter. Big deal.

That's the trick. Figure out what's really important and deal with the rest. You can't have your way all the time, and you're going to have to compromise. Don't take it personally.

Will Consulting Be Different?

Here's the clincher, though—where this stuff is important when it comes to consulting.

If you're an employee thinking about starting a consulting career, ask yourself what it is you really hate about your present job. Is it the capricious decisions imposed on you by a hostile and uncaring boss? Is it the lack of control in your life? Is it all the idiots you have to work with?

Now think a moment about how life will be different when you're a consultant. You won't have a boss. Depending on the work you do, you'll have perhaps six to ten bosses! From time to time, more than one boss will want something at the same time. After all, they don't exactly coordinate with one another because they represent different companies. In fact, they probably don't even know one another. If control is what you hate, you may have some difficulty with consulting. You'll be subject to a lot more strings al-

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

though clearly no one boss will ever wield the same influence over you.

That's certainly a key element in consulting. You may have as many orders coming down, but at least you can feel a better sense of control. You know that if one of the six people gets out of hand, you can always drop that contract. Well, you can drop it if you don't mind a little litigation and getting your reputation tarnished. I think it's a lot easier to quit as an employee than as a consultant. An employee is never finished with his or her task—the work by definition is not finite. Consultants, though, are hired to do a job, and shouldn't plan on quitting in the middle. Think about the difference. Would you feel more trapped as employee or consultant?

If you've been upset with your present employer for making you feel bad, you'll probably find that shorter term consulting contracts get you in and out before familiarity can breed contempt. I liked all the people in my present work group—until I got to know them. Then I began to dislike a lot of them. Gradually I've come to realize that each has his or her own problems, and as far as I can tell everyone else seems to have more problems than I have. That's a comforting thought. When you get to that stage, you can start to appreciate some of these characters for their unusual wit, marvelous chutzpa, or whatever. In short term contracts you won't be with people long enough (usually) for the friction to become really uncomfortable.

So, consulting can set you free from the proverbial yoke of oppression. But there are some psychological negatives as well.

There are three things I've found difficult to deal with as an independent consultant. The first problem is self-discipline. When you work for a company, you get paid day in and day out, pretty much without regard to the quality of work you do. Sure, if you do a lousy job, eventually you'll get canned. In the short run you'll probably get yelled out.

In independent consulting, though, doing a lousy job can cost you a contract overnight. You might not get paid for the work you've already done, and in the worst scenario you'll wind up in court. You've got to screw up big time as an employee to attract that sort of legal unpleasantness. So, if you're going to be independent, you've got to be good. Doing a good job means not only technical proficiency, but superior client relations as well.

The second psychological problem is pressure. If you work for a company, your boss knows, or should know, what you're tasked for. Since your time is finite, you can be over committed by your boss only to the degree that she or he fails to perceive reality. If your boss constantly promises others that you'll have a three week job done in two weeks, you're only 50% over committed. With a little negative reinforcement, you may even be able to break the boss of that habit.

In consulting, though, the diversity of clients means that inevitably six people will all want something done at the same time. Your time is still a finite resource, but the erratic demand for this time is certain to cause periods of over commitment. The more successful you are, the worse this problem gets. You may recall earlier discussions of the availability of billable hours. Because of office overhead and marketing activities, it's impossible to bill 40 hours per week and work a total of only 40 hours. So, unless you flat turn down work and leave clients in the lurch, you're going to become over committed at some time.

If you don't like pressure, you're going to hate independent consulting. What are you going to do two days before a deadline when another client calls and says his system has crashed? The crashee desperately needs your help, but if you miss the first client's deadline, you're dog meat. Oy. That's pressure.

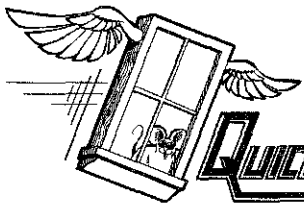
By the way, the smart consultant deals with this by increasing the reliability of the systems developed, and, taking a lesson from motorcycle safety courses (no, motorcycle safety is not an oxymoron), the consultant who survives learns to separate hazards. Only an idiot, or an inexperienced but enthusiastic consultant, would ever schedule two systems to go on-line in the same week. Separate your problems by scheduling them to avoid overlap.

The third psychological problem with consulting is money. Money is mostly a problem when you don't make any, which is usually all the time when you're starting your practice. Not making any money is very uncomfortable for most people.

One of my clients is of Irish ancestry and finds ethnic stereotyping a convenient way to deal with people. "Oh, your friend's Italian. Bet she's excitable, Rubenesque, and loves pizza, huh?" He's convinced I'm a penny pinching cheapskate because of my Scottish ancestry. What he calls cheap, I call prudent. I don't fly all over taking care of problems I can solve from the comfort of

SAVE HOURS OF VALUABLE PROGRAMMING TIME!

Now For QB 4.0



QuickWindows

A low cost Window Management Library for BASIC

- Create windows, mouse/kb driven pop-up and pull-down menus, and powerful data entry routines quickly and easily.
- Fast! Written entirely in assembly language
- Efficient! Uses less than 21K of object code
- Full mouse interface for Microsoft compatible mouse.
- Supports MDPA, CGA, and EGA automatically.

Comes complete with a library of over 60 functions, a fully-documented manual, many programming examples, and a demo/tutorial program to show you how to use each of the functions. All for only $79. Add $20 for the QuickWindows™ assembly language source code.

No royalties are required for using QuickWindows™ in your programs and not copy protected. 30-day money back guarantee. For IBM/True compatibles and Microsoft QuickBasic or BASCOM, DOS 2.0 or later.

Call now for your FREE demo disk!
401-274-5465
Visa, MC and Phone orders welcome.

Dealer discounts available.

Software Interphase INCORPORATED
5 Bradley Street, Suite 7
Providence, RI 02908-2304

Number 115 on the Reader Service Card

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

my office, using the telephone for both voice and data.

I don't rush disks out the door with overnight couriers when I know something will sit for a week. Hey, you want it fast, call my system and download it, or I'll add to your bill $10.75 for overnight mail. My experience is that if people wouldn't wait until the last damn minute, they wouldn't need things overnight anyway.

The real reason for my tight money management, though, is having been a starving consultant for a few years. I could take a client to a fancy restaurant one night, or feed my whole family for several days. It was not much of a choice. I felt obliged to do low-cost PR work and substitute effort and results for expensive flash.

There's a solution to the money problem, of course. Start out with lots of it. By the time you run out, you'll either be doing well as a consultant, or you will have given up. Alternatively, start with a long term consulting contract that provides some stability. It's always nice to know you'll be able to eat this month, and knowing that, the psychological problem of money disappears.

There are other psychological differences between being an employee and being a consultant, but all these issues revolve around your attitude. If you're truly unhappy in your present job and thinking about going into consulting, be sure to assess your present dissatisfaction. See if you can't do something to block out your problems at work, to make them lighter. When you feel that you've got your present job under control, then make the move to consulting. Don't rush into consulting to escape the turmoil of your present employment and thereby take all your bad attitudes with you.

A Billy Joel song talks about sleeping alone versus sleeping with someone else and concludes that either way you wake up with yourself. Recognize the differences between working as an employee and working as a consultant, but note that either way it's your attitude that will determine your success and happiness in your endeavors. We all wake up with ourselves, so it's best to get our attitudes and expectations in order, regardless of what we do for a living. When you've got your own head screwed on straight and can deal with the world, it's time to make your plans. If you truly know what you are about, both technically and psychologically, you're going to be a dynamite consultant. ♦

Send your comments or questions to

Paul A. Barkley
The Barbican Company
6845 Elm Street
Penthouse One
McLean, VA 22101

'Faster and better in no time'

Create better, faster, higher quality and easier to read progams in a fraction of the time. Let your system do the work for you. With 23 powerful, state of the art tools for the IBM PC and compatibles, both beginners and experts will find programming a breeze.

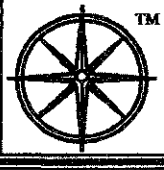
- Powerful, common user interace
- Interactive and batch execution
- Online documentation
- Self-explanatory error messages

C Programmer's Toolbox
Volumes I & II

- Monitor program/system execution
- Beautify program listings
- Determine program flow/critical path
- Trace/verify variable usage
- And more....

At $79.95 per volume or $130 for both with a 30 day trial, the Toolbox is simply the best value today. In addition, the documentation is packed with examples and tips on creating better programs.

Why waste time, call or write us today.



MMC AD Systems
Box 360845 Milpitas, California 95035
(408) 263-0781

Number 94 on the Reader Service Card

TRUE MULTITASKING
With
MultiDos Plus
"multitasking for the IBM-PC."

Ideal for developing applications in process control, data acquisition, communications, and other areas. Check these features which make **MultiDos Plus** an unbeatable value.

- Run up to 32 programs concurrently.
- Your software continues to run under DOS. No need to learn a new operating system.
- Use the compilers you already have. Supports software written in most languages.
- Operator commands to load/run programs, change priority, check program status, abort/suspend/resume programs.
- Programmatic interface via INT 15H for the following.
  - * Intertask message communication. Send/receive/check message present on 64 message queues.
  - * Task control by means of semaphores. Get/release/check semaphores.
  - * Change priority-256 priority levels.
  - * Suspend task for specified interval.
  - * Spawn and terminate external and internal tasks.
  - * Disable/enable multitasking.
  - * and more!
- Independent foreground/background displays.
- Access to DOS while applications are running.

Hardware/Software Requirements

IBM PC/XT/AT or true clone. Enough memory to hold **MultiDos Plus** (48 KB) and all your application programs. Also may need 4 or 16 KB memory for "hidden screens" for each active task. MS-DOS (or PC-DOS) 2.0 or later operating system.

only: $24.95 *OR* $99.95 with source code

Outside USA add $5.00 shipping and handling.
Visa and Mastercard orders only call toll-free: 1-800-872-4566, ext. 350., or send check or money order to:

NANOSOFT
13 Westfield Rd, Natick, MA 01760
MA orders add 5% sales tax.

Number 109 on the Reader Service Card

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

Write Mode 3 of the VGA

by
Michael Abrash

Well, the VGA is here, and it's a dandy, with 256 color capability (from 256K, no less), higher resolution, square pixels, readable registers (finally!), lots of new BIOS functions, better text, relatively good performance, and more. The VGA represents a whole new level of video capability, providing all the basics needed for decent business graphics for the first time in standard IBM graphics. The VGA provides reasonable backward compatibility with the CGA, MDA, and, most importantly, EGA. In short, the first video circuitry IBM has ever built into a motherboard is good, about as good as it could be given the lack of a built-in graphics processor and the constraints of backward compatibility.

For *PJ* readers, of course, the concern is how the VGA will affect their software development efforts. The answer: a great deal. All Micro Channel computers have the VGA built in; the PC/XT/AT version of the VGA, the Display Adapter, is priced at a fairly inexpensive $595; and third-party video companies are scrambling to be the first to clone the VGA. Between the benefits of the VGA and IBM's commitment to the new standard, the VGA should take off faster than even the EGA did.

So, readers, support the VGA in your development efforts—it's the hottest thing around right now. Besides, if you want your software to run in the Model 50 and up PS/2 computers, it *has* to run on the VGA, although the BIOS does a good job of masking the differences between the VGA and earlier adapters for older applications.

I'm not going to do a review/evaluation of the VGA; you'll have plenty of opportunities to read such articles in other magazines, and my theory is that *PJ* readers want practical tips and working code. With that in mind, I'm going to note a few points about identifying and programming the VGA and then revisit an old application in order to illuminate an unusual VGA programming feature, write mode 3.

Notes on Programming the VGA

The first point about the VGA is that while it's certainly different from earlier adapters, most old software should run on it. CGA and MDA software should run on the VGA to about the extent to which they run on the EGA, which is to say the VGA is BIOS-level compatible with the CGA and MDA. Most EGA software should run on the VGA so long as the BIOS is used whenever possible. While changing the hardware itself considerably, IBM managed to retain in the VGA the combination BIOS/register interface through which well-behaved application programs access the EGA. This doesn't mean that programs that load all the registers directly, or, indeed, touch any more registers than they absolutely must, will run; most VGA registers are, in fact, not EGA compatible. The EGA programming techniques I've discussed in *PJ* will work on the VGA. The rule of thumb: When in doubt, refer to the EGA and VGA technical reference material and make sure that the registers you're accessing are functionally the same.

VGA Reference Material

Which brings me to my next point: The VGA technical reference material is included in the *Technical Reference* manuals for the Micro Channel computers, including the *Model 50 and 60 Technical Reference* and the *Model 80 Technical Reference*. The Display Adapter technical reference material is available as a separately purchased supplement, the *Display Adapter Technical Reference* (part number S68X-2251-0). Oddly enough, the register descriptions in the *Display Adapter Technical Reference* and in the technical references for the Micro Channel computers are not identical although they seem to describe the same registers. It may be worth your while to buy both the *Display Adapter Technical Reference* and one of the Micro Channel computer technical references in order to have two descriptions of some registers.

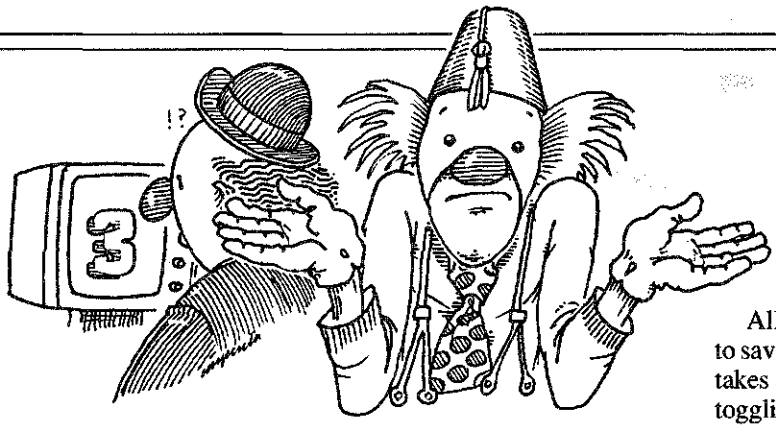
The *Display Adapter Technical Reference* includes a specification of the interface to the Display Adapter BIOS (which seems to be functionally pretty much the same as the VGA BIOS) as well as a register-level description of the Display Adapter. The technical references for Micro Channel computers do not contain any BIOS interface information; if you don't have the *Display Adapter Technical Reference*, you can get a specification of the entire IBM BIOS interface (including all IBM video BIOS functions) by purchasing IBM's *BIOS Interface Technical Reference* manual (part number S68X-2260-00).

Differences between the VGA and the Display Adapter

IBM notes a few significant differences between the Display Adapter and the VGA. (Incidentally, VGA stands for Video Graphics Array, which describes the chip on the Micro Channel motherboard. So far as I can determine, the same chip is on the Display Adapter—at least it has the same part number and does the same things—but it's easiest to refer to the motherboard video circuitry as the VGA and to the PC/XT/AT plug-in board as the Display Adapter.) I'll discuss these differences, which include vertical interrupt capability and monochrome text attributes, next.

Although the VGA can generate vertical interrupts of the same sort as the EGA does, the Display Adapter has no vertical interrupt capability. The reasoning behind this difference isn't clear, but I tested the Display Adapter and the vertical interrupt is indeed not available.

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED



On page 4-5 of the *Display Adapter Technical Reference*, there is a note to the effect that "the Personal System/2 Display Adapter displays a reverse video intensified character as white on white." From that, it's not clear exactly what the problem is, so I did some experimentation. So far as I can tell, when the Display Adapter is in mode 7, characters for which either the foreground or background attribute is 8 are displayed as white-on-white and are consequently not readable. The VGA, on the other hand, does display such text correctly. I'm not sure why this problem occurs or what a workaround might be; anyone out there have any suggestions?

By the way, the VGA is capable of driving either a color or a monochrome monitor automatically. (The monitors must be analog monitors compatible with IBM's PS/2 monitors, the 8503, 8512, 8513, and 8514.) There are no switches; the VGA IDs the monitor through the video connector. Color summing to gray scales is performed when driving a monochrome monitor. What's more, in Micro Channel computers, programs can select any mode directly through INT 10h, function 0; for example, a program can go from mode 3 (color text) to mode 7 (monochrome text) simply by performing

```
MOV AX,7
INT 10H
```

It is no longer necessary to fiddle with the equipment flag in order to switch between color and monochrome modes (in fact, the VGA BIOS changes the equipment flag as needed on mode sets), and the VGA by itself can act like any of the CGA, MDA, or EGA adapters at any time (although it can do this only serially; it can't emulate both at once since it is in truth only a single adapter).

The Display Adapter can drive either a monochrome or a color monitor and also features switchless installation. Switching between color and monochrome operation, however, still requires changing the equipment flag; unlike the VGA, the Display Adapter is constrained by the possible presence of other adapters in the system. Overall, IBM seems to have designed the BIOS and the VGA in such a way that application programs will run just as they would on earlier adapters in most cases.

One more note about programming the VGA: Don't use word OUTs to the Attribute Controller at port 3C0h. The EGA addresses the Attribute Controller at both 3C0h and 3C1h, so word OUTs to the Attribute Controller work despite its strange toggling nature. (See my article "Inside the EGA," *PJ*, Volume 5.1, Jan/Feb 1987, for a description of programming the Attribute Controller.) The VGA addresses the Attribute Controller at 3C0h only, meaning that the high byte of a value written to the VGA's Attribute Controller by a word OUT goes into the bit bucket. Word OUTs to the Sequencer, the CRT Controller, and the Graphics Controller seem to work fine on the VGA.

All VGA registers are readable, which makes it possible to save the video context (for example, in a TSR program that takes over the screen). While the VGA maintains the EGA's toggling Attribute Controller port operation on writes, it implements Attribute Controller reads in a sensible fashion. The Attribute Controller Index register is always readable at 3C0h, and the Attribute Controller Data register is always readable at 3C1h. Sadly, the need for EGA compatibility prevented IBM from making Attribute Controller writes operate equally sensibly.

There is one aspect of the VGA state that can't be read out and which consequently can't be saved on a context switch, and that's the state of the Attribute Controller index/data toggle. As a result, it seems advisable to disable interrupts while programming Attribute Controller registers, lest an interrupt come along and trigger a TSR that accesses the Attribute Controller. Better yet, use BIOS function to program Attribute Controller registers whenever possible. The palette RAM registers (Attribute Controller registers 0 through 0Fh) can be written (and read) through the BIOS, and these are usually the only Attribute Controller registers that application programs change.

Identifying the VGA

IBM has added a couple of functions to the VGA BIOS (and to the Model 30 BIOS as well) that make it much easier to identify installed adapters and their capabilities. Both are accessed through the standard INT 10h video interrupt.

Function 1Ah returns a display combination code indicating the type of the currently active display adapter and the alternate display adapter, if any. Function 1Ah isn't supported by earlier adapters, so the first test is whether function 1Ah is supported by the installed adapter. If, on return from a function 1Ah call, register AL is set to 1Ah, then function 1Ah is supported. If the function is supported, BL returns the active display code and BH returns the alternate display code. Function 1Ah can also be used to change the active and alternate display adapters.

Function 1Bh returns a buffer containing functionality/state information. This information includes the current display mode, the width of the screen, the display buffer size, other BIOS variables, the display combination code, the number of colors supported by the current mode, and font information. Best of all, it includes information about the video modes supported by the active adapter. I haven't had a chance to dig into the functionality/state information yet, but it looks like a gold mine. If this function had been available since the MDA and CGA, the world of IBM graphics would be a lot less confusing.

Write Mode 3

The MDA offered only text mode, so it was easy to program. The CGA had a couple of graphics modes, and while the bit

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

map organization in these modes was somewhat complex, at least the bit map was directly addressed as one linear block of memory. The EGA added the complications of three write modes and two read modes, all across four planes. The VGA supports all the modes of the earlier adapters and adds one new write mode to the list, write mode 3.

Write mode 3 is strange indeed, and its use is not immediately obvious. In write mode 3, set /reset is automatically enabled for all four planes (the Enable Set/Reset register is ignored). The CPU data byte is rotated and then ANDed with the contents of the Bit Mask register, and the result of this operation is used as the contents of the Bit Mask register alone would normally be used. (If this is *Greek* to you, I suggest you read my earlier four articles on graphics in *PJ*, Volumes 5.1 through 5.4. There's no way to understand write mode 3 without understanding write mode 0 first.)

That's what write mode 3 does—but what is it for? I'm not certain what IBM had in mind, but the best applications for write mode 3 I've been able to think of to date have to do with bit-mapped text applications. Write mode 3 seems reasonably well-suited to drawing large quantities of graphics-mode text quickly without wiping out the background in the process.

Listing 1 is a modification of the code presented in my article "EGA Data Control" (*PJ*, Volume 5.3, May/June 1987). That code used the data rotate and bit mask features of the EGA to draw bit-mapped text in write mode 0. Listing 1 uses write mode 3 to draw bit-mapped text, and in the process gains the important benefit of preserving the background into which the text is being drawn. Where the original text-drawing code drew the entire character box for each character, with 0 bits in the font pattern causing a black box to appear around each character, the code in Listing 1 affects display memory only when 1 bits in the font pattern are drawn. As a result, the characters appear to be painted into the background, rather than over it. Another advantage of the code in Listing 1 is that the characters can be drawn in any of the 16 available colors.

The key to understanding Listing 1 is understanding the effect of ANDing the rotated CPU data with the contents of the Bit Mask register. The CPU data is the pattern for the character to be drawn, with bits equal to 1 indicating where character pixels are to appear. The Data Rotate register is set to rotate the CPU data to pixel-align it since without rotation characters could be drawn only on byte boundaries. At the same time, the Bit Mask register is set to allow the CPU to modify only that portion of the display memory byte accessed that the pixel-aligned character falls in so that other characters and/or graphics data won't be wiped out. The result of ANDing the rotated CPU data byte and the contents of the Bit Mask register is a bit mask that allows only the bits equal to 1 in the original character pattern (rotated and masked to provide pixel alignment) to be modified by the CPU; all other bits come straight from the latches. The latches should have previously been loaded from the target address, so the effect of the ultimate bit mask value is to allow the CPU to modify only those pixels in display memory that correspond to the bits equal to 1 in that part of the pixel-aligned character that falls in the currently addressed byte. The color of the pixels set by the CPU is determined by the contents of the Set/Reset register.

Whew. It sounds complex, but given an understanding of what the data rotator, set /reset, and the bit mask do, it's not that bad. One good way to make sense of it is to refer to the original text-drawing program in *PJ* 5.3, and then see how Listing 1 differs from that program.

LEARN about . . .

Neural Nets

A major upheaval in thinking about artificial intelligence is underway. Interest is accelerating in the "neural net" approach, which is loosely based on human brain research. A tutorial introduction to this exciting new technology is now available in a model called Netwurkz (tm).

Netwurkz eliminates the complicated math found in "unsupervised learning" so that underlying principles may be grasped quickly. The network "learns" when you add new DATA statements (interconnects), or modify existing ones. A trace on/off feature prints the neural net computation in progress. List based internal representation yields speed.

Netwurkz is intended for people not familiar with neural modeling who want to maintain or broaden their technical skills and background. Source for all neural net software and an object-only copy of the PL/D compiler are included

Compilers

PL/D (tm) is a new system language that contains documentation so definitive that your understanding of all languages is enhanced. Like other languages, PL/D is described in narrative form, alphabetically by token and via flow diagrams. However, PL/D also comes with self-compiling source! If you have a question after reading the manual, you can look at the source. If you do not accept the answer, you can change it by modifying the code.

What you learn from PL/D also applies to all compilers: symbol table organization, macro handling, nested structure implementation, cross reference listing, code generation, etc. You will study an unusually efficient implementation: PL/D is 2-3 times faster than the recent "fast" C compilers.

If you are a bright self-sufficient type who wants to master your development tools, PL/D is for you.

PC or compatible with 256K and DOS 2.0 or above and 5.25 inch diskette required.

DAIR Computer Systems 3440 Kenneth Drive, Palo Alto CA 94303 (415) 494-7081

Name: _____
Company: _____
Address: _____
City, State, Zip: _____
Phone: _____
VISA, M/C: _____ Expire: _____

Netwurkz: neural net demo	$79.95	
PL/D: fast self-compiles	$124.95	
Both of above	$154.95	
Sales Tax, CA only		
Ship overseas	$10.00	
TOTAL		

Want to know more? (415) 494-7081

Available Now

Number 5 on the Reader Service Card

Software Developers

We need your program!

Do you have a program that's good enough to sell, but don't want the problems or financial risk of producing, typesetting, printing, packaging, warehousing, marketing, distributing and supporting a product?

Why start your own software house?
We've done it for you!

Merlin Publishing Group
is now accepting submissions of micro-computer software for publication.

You get:
- initial cash payments
- generous royalties
- to spend your time programming

But we can't help you if you don't submit. Call or write today for our submission guideline kit.

MERLIN PUBLISHING GROUP ™

1240 Johnsons Ferry Place, Suite A10
Marietta, GA 30068
(404) 977-6034

P. S. See us in Atlanta at COMDEX-Spring '88

Number 141 on the Reader Service Card

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

It's worth noting that the same effect generated by Listing 1 could have been accomplished without write mode 3, and at relatively little performance cost. Write mode 0 could have been used instead. Rather than let write mode 3 rotate the CPU data and AND it with the contents of the Bit Mask register, the CPU could simply have rotated the CPU data directly and ANDed it with the value destined for the Bit Mask register and then set the Bit Mask register to the resulting value. Additionally, set/reset could simply be enabled for all planes, emulating what write mode 3 does to provide pixel colors. While this arrangement would have required a bit more bit manipulation by the CPU and would have required CPU rotation, which is slower than the VGA's rotator, it would have saved setting the Data Rotate register and actually might even have been a faster routine overall.

Write mode 3 could be considerably more useful for drawing large blocks of text. For example, suppose that we were to draw a line of 8-dot-wide, bit-mapped text 40 characters long. It would be very possible to set up the bit mask and data rotation as appropriate for the left portion of each bit-aligned character (the portion of each character to the left of the byte boundary) and then draw the left portions only of all 40 characters in write mode 3. Then the bit mask could be set up for the right portion of each character, and the right portions of all 40 characters could be drawn. The VGA's fast rotator would be used to do all rotation, and the only OUTs required would be those required to set the bit mask and data rotation. This technique could well outperform single-character bit-mapped text drivers such as the one in Listing 1 by a significant margin. Listing 2 illustrates one implementation of such an approach. Incidentally, note the use of the 8 x 14 ROM font in Listing 2, rather than the 8 x 8 ROM font used in Listing 1. There is also an 8 x 16 font stored in ROM, along with the tables used to alter the 8 x 14 and 8 x 16 ROM fonts into 9 x 14 and 9 x 16 fonts.

It's likely that there's a better application for write mode 3 than the sort of bit-mapped text handling I've discussed. If you think of such an application, please write and let *PJ*'s readers know about it.

Preserving Reserved Bits

Note that the code in Listing 1 uses the readable register feature of the VGA to preserve reserved bits and bits other than those being modified. The EGA has no readable registers, and so it was necessary to set all bits in a register whenever that register was modified. The VGA makes it possible to change only those bits of immediate interest, and this procedure is highly recommended since IBM (or clone manufacturers) may well use some of those reserved bits in the future.

In Conclusion

The VGA is a complex, powerful, fascinating video standard. There's a heck of a lot to learn, and knowledge is the key to high-performance VGA programs. One area I'll make it a point to cover soon is 256 color mode, another is the ability to select colors from a set of 256K, and yet another is text mode font handling. And then there's the new two-color high-resolution mode, the pel panning compatibility feature, the high-bandwidth blanking bit, the . . . . Yes indeed, there's a lot to cover next time. See you then. ♦

Michael Abrash is an Engineering Fellow working on advanced graphics projects for Video Seven, Inc., in Fremont, CA.

Code follows.

NEW!
FASTER THAN EVER!
DeSmet C V3.0

FASTER C DEVELOPMENT

Invoke the DeSmet C compiler from the SEE[tm] full screen editor and the first error will return you immediately to SEE at the error line with the error message displayed.

FASTER COMPILATION

When you don't use inline assembly code or don't want to see the ASM88 output, the V3.0 compiler produces object code directly - making DeSmet C up to ***twice as fast*** as before.

PLUS EXPANDED STANDARD LIBRARY

Networking, path, file, time, enhanced string functions, environment support now included.

FULL FEATURES WITH EVERY PACKAGE
— ONLY $109 —

C Compiler, Assembler, Binder, Librarian, Execution Profiler, Overlays, 8087 and S/W Floating Point, Full STDIO Library and Full Screen Editor (SEE). Debugger and Large Case options available at $50 each.

C Ware Corporation
P.O. Box 428, Paso Robles, CA 93447
Phone: (805) 239-4620 **Telex:** 358185 **BBS:** (805) 239-4834
We accept **VISA, MC & AMEX.** Call now and we'll ship today.
Street Address: 945 Spring #14, Paso Robles, CA 93446

Number **38** on the Reader Service Card

Introducing

The only code-intensive publication on C for MS-DOS® systems.

- Code-intensive: pages of clear, fully-commented and portable C. Extensive textual support for the code.
- In-depth coverage of algorithms: detailed explanations presented from different perspectives by several authors.

Call or write today for your subscription.

1 year (4 issues) $21. **2 years (8 issues) $38.**

The C Gazette

1341 Ocean Ave. #257 Santa Monica, CA 90401
(213) 473-7414

Number **144** on the Reader Service Card

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

Code from "Write Mode 3..." by Michael Abrash

1

```
Listing 1

;Program to illustrate operation of write mode 3 of the VGA.
;  Draws 8x8 characters at arbitrary locations without disturbing
;  the background, using VGA's 8x8 ROM font.  Designed
;  for use with modes 0Dh, 0Eh, 0Fh, 10h, and 12h.
; Runs only on VGAs (in Models 50 & up and IBM Display Adapter
;  and 100% compatibles).
; Assembled with MASM 4.0, linked with LINK 3.51.
; By Michael Abrash, 9/1/87.
;
stack   segment para stack 'STACK'
        db      512 dup(?)
stack   ends
;
VGA_VIDEO_SEGMENT       equ     0a000h  ;VGA display memory segment
SCREEN_WIDTH_IN_BYTES   equ     044ah   ;offset of BIOS variable
FONT_CHARACTER_SIZE     equ     8       ;# bytes in each font char
;
; VGA register equates.
;
SC_INDEX        equ     3c4h    ;SC index register
SC_MAP_MASK     equ     2       ;SC map mask register index
GC_INDEX        equ     3ceh    ;GC index register
GC_SET_RESET    equ     0       ;GC set/reset register index
GC_ENABLE_SET_RESET equ 1       ;GC enable set/reset register index
GC_ROTATE       equ     3       ;GC data rotate/logical function
                                ; register index
GC_MODE         equ     5       ;GC Mode register
GC_BIT_MASK     equ     8       ;GC bit mask register index
;
dseg    segment para common 'DATA'
TEST_TEXT_ROW   equ     69      ;row to display test text at
TEST_TEXT_COL   equ     17      ;column to display test text at
TEST_TEXT_WIDTH equ     8       ;width of a character in pixels
TestString      label   byte
        db      'Hello, world!',0       ;test string to print.
FontPointer     dd      ?               ;font offset
dseg    ends
;
cseg    segment para public 'CODE'
        assume  cs:cseg, ds:dseg
start   proc    near
        mov     ax,dseg
        mov     ds,ax
;
; Select 640x480 graphics mode.
;
        mov     ax,012h
        int     10h
;
; Set the screen to all blue, using the readability of VGA registers
; to preserve reserved bits.
;
        mov     dx,GC_INDEX
        mov     al,GC_SET_RESET
        out     dx,al
        inc     dx
        in      al,dx
        and     al,0f0h
        or      al,1            ;blue plane only set, others reset
        out     dx,al
        dec     dx
        mov     al,GC_ENABLE_SET_RESET
        out     dx,al
        inc     dx
        in      al,dx
        and     al,0f0h
        or      al,0fh          ;enable set/reset for all planes
        out     dx,al
        mov     dx,VGA_VIDEO_SEGMENT
        mov     es,dx           ;point to display memory
        mov     di,0
        mov     cx,8000h        ;fill all 32k words
        mov     ax,0ffffh       ;because of set/reset, the value
                                ; written actually doesn't matter
        rep stosw               ;fill with blue
```

2

```
; Set driver to use the 8x8 font.
;
        mov     ah,11h  ;VGA BIOS character generator function,
        mov     al,30h  ; return info subfunction
        mov     bh,3    ;get 8x8 font pointer
        int     10h
        call    SelectFont
;
; Print the test string, cycling through colors.
;
        mov     si,offset TestString
        mov     bx,TEST_TEXT_ROW
        mov     cx,TEST_TEXT_COL
        mov     ah,0            ;start with color 0
StringOutLoop:
        lodsb
        and     al,al
        jz      StringOutDone
        push    ax              ;preserve color
        call    DrawChar
        pop     ax              ;restore color
        inc     ah              ;next color
        and     ah,0fh          ;colors range from 0 to 15
        add     cx,TEST_TEXT_WIDTH
        jmp     StringOutLoop
StringOutDone:
;
; Wait for a key, then set to text mode & end.
;
        mov     ah,1
        int     21h     ;wait for a key
        mov     ax,3
        int     10h     ;restore text mode
;
; Exit to DOS.
;
        mov     ah,4ch
        int     21h
Start   endp
;
; Subroutine to draw a text character in a linear graphics mode
;  (0Dh, 0Eh, 0Fh, 010h, 012h). Background around the pixels that
;  make up the character is preserved.
; Font used should be pointed to by FontPointer.
;
; Input:
;  AL = character to draw
;  AH = color to draw character in (0-15)
;  BX = row to draw text character at
;  CX = column to draw text character at
;
;  Forces ALU function to "move".
;  Forces write mode 3.
;
DrawChar        proc    near
        push    ax
        push    bx
        push    cx
        push    dx
        push    si
        push    di
        push    bp
        push    ds

        push    ax      ;preserve character to draw in AL
;
; Set up set/reset to produce character color, using the readability
; of VGA register to preserve the setting of reserved bits 7-4.
;
        mov     dx,GC_INDEX
        mov     al,GC_SET_RESET
        out     dx,al
        inc     dx
        in      al,dx
        and     al,0f0h
        and     ah,0fh
        or      al,ah
        out     dx,al
```

Code continues...

PRODUCED 2005 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

Announcing WKS LIBRARY

NEW!

The Lotus "Wrap-Around" for C Programs

Now you can write and read Lotus worksheets directly from a C program!

WKS LIBRARY lets you:

- avoid time-consuming file translation steps
- control the execution of 1-2-3 from inside your application
- use Lotus as a data entry screen in your C program
- generate "live" financial statements with formulas for totals

Feature this:

- reads and writes .WKS, .WK1 and .WR1 worksheets
- writes integers, floats, strings, formulas, macros and dates using *wprintf()*
- reads using *wscanf()*, converting column contents to C variables according to format specs
- has low-level functions for manipulating individual cells
- provides Lotus control functions such as: formats, column widths, initial cursor position, range names, cell protection
- supports Lattice, Microsoft & Turbo compilers for DOS, plus most UNIX environments

Source Code provided **No Royalties on executable programs**

Only $89
(includes free 800-line support)

VISA MasterCard

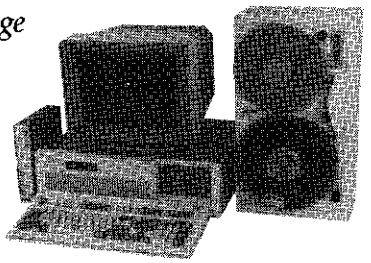
ORDER TODAY!
Toll-free
(800) 367-9882

Tenon Software, Inc.
1980 - 112th NE, #250, Bellevue, WA 98004
(206) 453-1914 (in Washington state)

Number **163** on the Reader Service Card

9-Track Tape Subsystem

The Data Interchange Solution You've Been Waiting For!



Qualstar's ½" 9-track Ministreamers™ bring full ANSI data interchange capability to your PC or Macintosh™ computer system.

With 9-track tape, you are free to exchange data files with any mainframe or minicomputer in the world.

Our affordable Ministreamers come in both 7" and 10½" versions and use less desk space than an ordinary sheet of paper. They can provide 1600 thru 6250 BPI capability and may be used for disk backup as well as data interchange. Complete subsystem prices, including software, start as low as $2,495 for 7" units and at $3,670 for 10½" units.

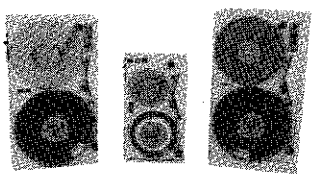
Qualstar has become the market leader in desk-top 9-track systems for a good reason; our tape drives have established an outstanding record for reliability and low cost of ownership.

Discover the many advantages 9-track tape has over other Micro/Mainframe links.

Call us today!

QUALSTAR®

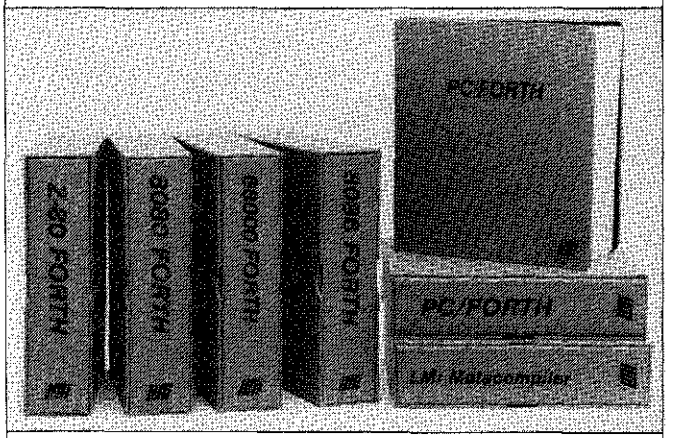
9621 Irondale Avenue,
Chatsworth, CA 91311
Telephone: (818) 882-5822



Macintosh is a trademark of Apple Computer, Inc.

Number **97** on the Reader Service Card

TOTAL CONTROL with *LMI FORTH*™



For Programming Professionals:

an expanding family of compatible, high-performance, Forth-83 Standard compilers for microcomputers

For Development: Interactive Forth-83 Interpreter/Compilers

- 16-bit and 32-bit implementations
- Full screen editor and assembler
- Uses standard operating system files
- 400 page manual written in plain English
- Options include software floating point, arithmetic coprocessor support, symbolic debugger, native code compilers, and graphics support

For Applications: Forth-83 Metacompiler

- Unique table-driven multi-pass Forth compiler
- Compiles compact ROMable or disk-based applications
- Excellent error handling
- Produces headerless code, compiles from intermediate states, and performs conditional compilation
- Cross-compiles to 8080, Z-80, 8086, 68000, 6502, 8051, 8096, 1802, and 6303
- No license fee or royalty for compiled applications

For Speed: CForth Application Compiler

- Translates "high-level" Forth into in-line, optimized machine code
- Can generate ROMable code

Support Services for registered users:

- Technical Assistance Hotline
- Periodic newsletters and low-cost updates
- Bulletin Board System

Call or write for detailed product information and prices. Consulting and Educational Services available by special arrangement.

Laboratory Microsystems Incorporated
Post Office Box 10430, Marina del Rey, CA 90295
Phone credit card orders to: ***(213) 306-7412***

Overseas Distributors.
Germany: Forth-Systeme Angelika Flesch, Titisee-Neustadt, 7651-1665
UK: System Science Ltd., London, 01-248 0962
France: Micro-Sigma S.A.R.L., Paris, (1) 42.65.95.16
Japan: Southern Pacific Ltd., Yokohama, 045-314-9514
Australia: Wave-onic Associates, Wilson, W.A., (09) 451-2946

Number **63** on the Reader Service Card

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

3

```
;
; Select write mode 3, using the readability of VGA registers
; to leave bits other than the write mode bits unchanged.
;
        mov     dx,GC_INDEX
        mov     al,GC_MODE
        out     dx,al
        inc     dx
        in      al,dx
        or      al,3
        out     dx,al
;
; Set DS:SI to point to font and ES to point to display memory.
;
        lds     si,[FontPointer]        ;point to font
        mov     dx,VGA_VIDEO_SEGMENT
        mov     es,dx                   ;point to display memory
;
; Calculate screen address of byte character starts in.
;
        pop     ax      ;get back character to draw in AL

        push    ds      ;point to BIOS data segment
        sub     dx,dx
        mov     ds,dx
        xchg    ax,bx
        mov     di,ds:[SCREEN_WIDTH_IN_BYTES]   ;retrieve BIOS
                                                ; screen width
        pop     ds
        mul     di      ;calculate offset of start of row
        push    di      ;set aside screen width
        mov     di,cx   ;set aside the column
        and     cl,0111b ;keep only the column in-byte address
        shr     di,1
        shr     di,1
        shr     di,1    ;divide column by 8 to make a byte address
        add     di,ax   ;and point to byte
;
; Calculate font address of character.
;
        sub     bh,bh
        shl     bx,1    ;assumes 8 bytes per character; use
        shl     bx,1    ; a multiply otherwise
        shl     bx,1    ;offset in font of character
        add     si,bx   ;offset in font segment of character
;
; Set up the GC rotation. In write mode 3, this is the rotation
; of CPU data before it is ANDed with the Bit Mask register to
; form the bit mask. Force the ALU function to "move". Uses the
; readability of VGA registers to leave reserved bits unchanged.
;
        mov     dx,GC_INDEX
        mov     al,GC_ROTATE
        out     dx,al
        inc     dx
        in      al,dx
        and     al,0e0h
        or      al,cl
        out     dx,al
;
; Set up BH as bit mask for left half, BL as rotation for right half.
;
        mov     bx,0ffffh
        shr     bh,cl
        neg     cl
        add     cl,8
        shl     bl,cl
;
; Draw the character, left half first, then right half in the
; succeeding byte, using the data rotation to position the character
; across the byte boundary and then using write mode 3 to combine the
; character data with the bit mask to allow the set/reset value (the
; character color) through only for the proper portion (where the
; font bits for the character are 1) of the character for each byte.
; Wherever the font bits for the character are 0, the background
; color is preserved.
; Does not check for case where character is byte-aligned and
; no rotation and only one write is required.
```

4

```
;
        mov     bp,FONT_CHARACTER_SIZE
        mov     dx,GC_INDEX
        pop     cx      ;get back screen width
        dec     cx
        dec     cx      ; -2 because do two bytes for each char
CharacterLoop:
;
; Set the bit mask for the left half of the character.
;
        mov     al,GC_BIT_MASK
        mov     ah,bh
        out     dx,ax
;
; Get the next character byte & write it to display memory.
; (Left half of character.)
;
        mov     al,[si]         ;get character byte
        mov     ah,es:[di]      ;load latches
        stosb                   ;write character byte
;
; Set the bit mask for the right half of the character.
;
        mov     al,GC_BIT_MASK
        mov     ah,bl
        out     dx,ax
;
; Get the character byte again & write it to display memory.
; (Right half of character.)
;
        lodsb                   ;get character byte
        mov     ah,es:[di]      ;load latches
        stosb                   ;write character byte
;
; Point to next line of character in display memory.
;
        add     di,cx
;
        dec     bp
        jnz     CharacterLoop
;
        pop     ds
        pop     bp
        pop     di
        pop     si
        pop     dx
        pop     cx
        pop     bx
        pop     ax
        ret
DrawChar        endp
;
; Set the pointer to the font to draw from to ES:BP.
;
SelectFont      proc    near
        mov     word ptr [FontPointer],bp       ;save pointer
        mov     word ptr [FontPointer+2],es
        ret
SelectFont      endp
;
cseg    ends
        end     start
```

Listing 2

```
; Program to illustrate high-speed text-drawing operation of
;  write mode 3 of the VGA.
;  Draws a string of 8x14 characters at arbitrary locations
;  without disturbing the background, using VGA's 8x14 ROM font.
;  Designed for use with modes 0Dh, 0Eh, 0Fh, 10h, and 12h.
; Runs only on VGAs (in Models 50 & up and IBM Display Adapter
;  and 100% compatibles).
; Assembled with MASM 4.0, linked with LINK 3.51.
; By Michael Abrash, 9/1/87.
;
stack   segment para stack 'STACK'
        db      512 dup(?)
```

Code continues...

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

EVEN MORE POWER AND FLEXIBILITY

BRIEF 2.0

Users and industry press alike have unanimously proclaimed BRIEF as the best program editor available today. Now, the best gets better, with the release of BRIEF 2.0.

Straight from the box, BRIEF offers an exceptional range of features. Many users find that BRIEF is the only editor they'll ever need, with features like real, multi-level Undo, flexible windowing and unlimited file size. But BRIEF has tremendous hidden power in its exclusive macro language. With it, you can turn BRIEF into your own custom editor containing the commands and features you desire. It's fast and easy.

Jerry Pournelle, columnist for BYTE magazine summed it all up by saying BRIEF is, "Recommended. If you need a general purpose PC programming editor, look no further." His point of view has been affirmed by rave reviews in C JOURNAL, COMPUTER LANGUAGE, DR. DOBB'S JOURNAL, DATA BASED ADVISOR, INFOWORLD AND PC MAGAZINE.

One user stated, "BRIEF is one of the few pieces of software that I would dare call a masterpiece." Order BRIEF now and find out why. BRIEF 2.0 is just $195. If you already own BRIEF, call for upgrade information.

TO ORDER CALL: 1-800-821-2492
(in MA call 617-337-6963)

As always, BRIEF comes with a 30 day money-back satisfaction guarantee.

Solution Systems™
541 Main Street
Suite 410
So. Weymouth, MA 02190
(617) 337-6963

Look at these BRIEF 2.0 enhancements!

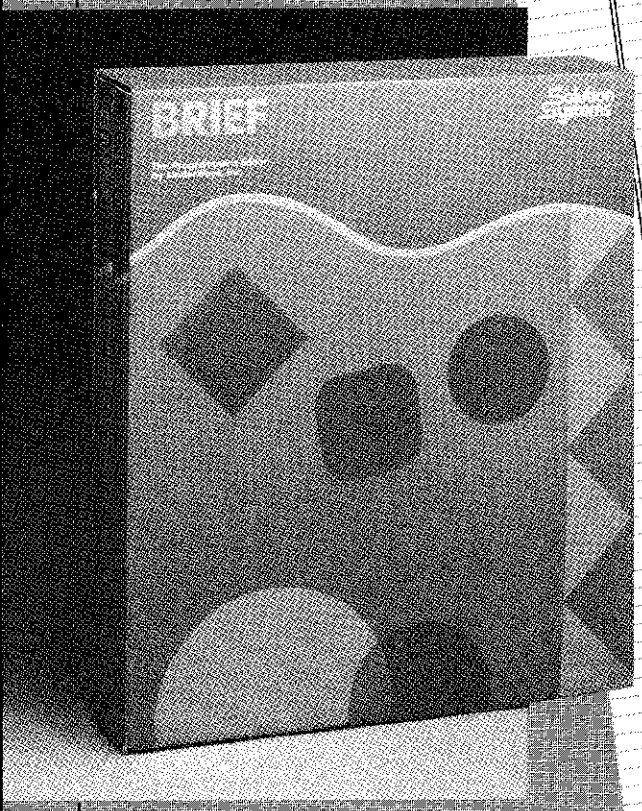
Main Features:

- All new documentation with tutorials on basic editing, regular expressions **and** the BRIEF Macro Language.
- Setup program for easy installation and configuration. (Requires no knowledge of the macro language)
- Increased speed for sophisticated operations like Undo and Regular Expression Search.
- Expanded regular expressions, with matching over line boundaries.
- More block types, with marking by character, line or column.
- Command line editing (move cursor, add and delete characters, specify command parameters).
- Support for more programming languages.
- Optional borderless windows.
- Enhanced large display support, including wider displays.
- Reconfigurable indenting for C files (supports most indenting styles).

Plus the basic features that made BRIEF SO popular!

Basic Features:

- Full multi-level Undo
- Windows
- Edit many files at once
- File size limited only by disk space
- Automatic language sensitive indentation



Requires an IBM PC or compatible with at least 192K RAM.
BRIEF is a trademark of UnderWare Inc.
Solution Systems is a trademark of Solution Systems.

Number 47 on the Reader Service Card.

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

Abrash Code, Continued

5

```
stack   ends
;
VGA_VIDEO_SEGMENT       equ     0a000h  ;VGA display memory segment
SCREEN_WIDTH_IN_BYTES   equ     044ah   ;offset of BIOS variable
FONT_CHARACTER_SIZE     equ     14      ;# bytes in each font char
;
; VGA register equates.
;
SC_INDEX        equ     3c4h    ;SC index register
SC_MAP_MASK     equ     2       ;SC map mask register index
GC_INDEX        equ     3ceh    ;GC index register
GC_SET_RESET    equ     0       ;GC set/reset register index
GC_ENABLE_SET_RESET equ 1       ;GC enable set/reset register index
GC_ROTATE       equ     3       ;GC data rotate/logical function
                                ; register index
GC_MODE         equ     5       ;GC Mode register
GC_BIT_MASK     equ     8       ;GC bit mask register index
;
dseg    segment para common 'DATA'
TEST_TEXT_ROW   equ     69      ;row to display test text at
TEST_TEXT_COL   equ     17      ;column to display test text at
TEST_TEXT_COLOR equ     0fh     ;high intensity white
TestString      label   byte
        db      'Hello, world!',0       ;test string to print.
FontPointer     dd      ?               ;font offset
dseg    ends
;
cseg    segment para public 'CODE'
        assume  cs:cseg, ds:dseg
start   proc    near
        mov     ax,dseg
        mov     ds,ax
;
; Select 640x480 graphics mode.
;
        mov     ax,012h
        int     10h
;
; Set the screen to all blue, using the readability of VGA registers
; to preserve reserved bits.
;
        mov     dx,GC_INDEX
        mov     al,GC_SET_RESET
        out     dx,al
        inc     dx
        in      al,dx
        and     al,0f0h
        or      al,1            ;blue plane only set, others reset
        out     dx,al
        dec     dx
        mov     al,GC_ENABLE_SET_RESET
        out     dx,al
        inc     dx
        in      al,dx
        and     al,0f0h
        or      al,0fh          ;enable set/reset for all planes
        out     dx,al
        mov     dx,VGA_VIDEO_SEGMENT
        mov     es,dx           ;point to display memory
        mov     di,0
        mov     cx,8000h        ;fill all 32k words
        mov     ax,0ffffh       ;because of set/reset, the value
                                ; written actually doesn't matter
        rep stosw               ;fill with blue
;
; Set driver to use the 8x14 font.
;
        mov     ah,11h  ;VGA BIOS character generator function,
        mov     al,30h  ; return info subfunction
        mov     bh,2    ;get 8x14 font pointer
        int     10h
        call    SelectFont
;
; Print the test string.
;
        mov     si,offset TestString
        mov     bx,TEST_TEXT_ROW
        mov     cx,TEST_TEXT_COL
```

6

```
        mov     ah,TEST_TEXT_COLOR
        call    DrawString
;
; Wait for a key, then set to text mode & end.
;
        mov     ah,1
        int     21h     ;wait for a key
        mov     ax,3
        int     10h     ;restore text mode
;
; Exit to DOS.
;
        mov     ah,4ch
        int     21h
Start   endp
;
; Subroutine to draw a text string left-to-right in a linear
;  graphics mode (0Dh, 0Eh, 0Fh, 010h, 012h) with 8-dot-wide
;  characters. Background around the pixels that make up the
;  characters is preserved.
; Font used should be pointed to by FontPointer.
;
; Input:
;  AH = color to draw string in
;  BX = row to draw string on
;  CX = column to start string at
;  DS:SI = string to draw
;
;  Forces ALU function to "move".
;  Forces write mode 3.
;
DrawString      proc    near
        push    ax
        push    bx
        push    cx
        push    dx
        push    si
        push    di
        push    bp
        push    ds
;
; Set up set/reset to produce character color, using the readability
; of VGA register to preserve the setting of reserved bits 7-4.
;
        mov     dx,GC_INDEX
        mov     al,GC_SET_RESET
        out     dx,al
        inc     dx
        in      al,dx
        and     al,0f0h
        and     ah,0fh
        or      al,ah
        out     dx,al
;
; Select write mode 3, using the readability of VGA registers
; to leave bits other than the write mode bits unchanged.
;
        mov     dx,GC_INDEX
        mov     al,GC_MODE
        out     dx,al
        inc     dx
        in      al,dx
        or      al,3
        out     dx,al
        mov     dx,VGA_VIDEO_SEGMENT
        mov     es,dx                   ;point to display memory
;
; Calculate screen address of byte character starts in.
;
        push    ds      ;point to BIOS data segment
        sub     dx,dx
        mov     ds,dx
        mov     di,ds:[SCREEN_WIDTH_IN_BYTES]   ;retrieve BIOS
                                                ; screen width
        pop     ds
        mov     ax,bx   ;row
        mul     di      ;calculate offset of start of row
        push    di      ;set aside screen width
```

Code continues...

PRODUCED 2002 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

Make your C language programs memory resident with

DMS RESIDENT-C

Lattice		Microsoft
"hot-key"		enable
$79.95		$149.95 w/source

Make your assembler programs memory resident with

DMS RESIDENT-ASM

"hot key"		enable
$79.95		$149.95 w/source

American Software International
P.O. Box 523
Windsor, CT 06095-9998
(203) 688-5054

Number **151** on the Reader Service Card

EARLY BIRD SPECIAL -- Buy before March 1 for these low, low prices!!

The **A**spen **S**ystems **S**ubroutine **E**ditor (**ASE**) you can call from your programs. Design your own screen layouts - - update in several windows simultaneously. **ASE** is:

- ★ **TRANSPORTABLE** - can be configured for any keyboard and almost any computer or language under MSDOS.
- ★ **VERSATILE** - customize input for any application.
- ★ **COMPLETE** - with full screen edit capabilities and a wide variety of automatic conversions.
- ★ **EASY TO USE** - data and screen layouts described in a single map.
- ★ **FULLY DOCUMENTED** - including examples in BASIC, PASCAL, FORTRAN, C, and even COBOL.
- ★ **AFFORDABLE-**
  ASE sale priced at..... **$79**
  DEMO (See what ASE can do for you!)..... **$ 3**

The **A**spen **S**ystems **S**ubroutine **P**ackage (**ASP**) - the companion package for **ASE** - with many functions unavailable or difficult to perform in high level languages. Like **ASE, ASP** is:

TRANSPORTABLE ★ FULLY DOCUMENTED
EFFICIENT ★ AFFORDABLE

ASP includes 150+ subroutines: conversions; sorts; string, bit, date/time functions; decimal arithmetic, and **MORE.**

ASP sale priced at..... **$99**

ASE and **ASP** include support and demo programs.

Prices are PPD (continental USA).
Colorado Residents add 3%.

Aspen Systems

P. O. Box 1163
Grand Junction, CO 81502
(303) 245-3262
VISA/MasterCard accepted

CP/M and MS-DOS are trademarks of Digital Research and Microsoft, respectively.

Number **86** on the Reader Service Card

Unbelievable!

SOURCER™

Creates commented source code and listings from memory, .COM files or .EXE files

- **CLARIFY UNDOCUMENTED CODE**
- **EASILY MODIFY PROGRAMS**
- **RECREATE LOST SOURCE CODE**

SOURCER™ creates detailed commented listings and source code directly suitable for assembly. Built in data analyzer and simulator resolves multiple data segments and provides detailed comments on interrupts and subfunctions, I/O ports and much more. Determines all necessary assembler directives. Complete support for 8088 through 80286, V20/V30, 8087 and 80287 instruction sets. No other product comes close to the output quality of SOURCER.

BIOS SOURCE

PS/2 ▪ AT ▪ XT ▪ PC ▪ Clones

- **CHANGE & ADD FEATURES**
- **CLARIFIES BIOS INTERFACES**
- **SPECIFIC TO YOUR MACHINE**

The bios pre-processor to SOURCER™ provides the first means to obtain accurate legal source listings for any bios! Identifies entry points with full explanations. Resolves PS/2's multiple jumps for improved clarity. Provides highly descriptive data labels such as "video_mode" and "keybd_q_head," and much more. Fully automatic.

SOURCER™	$ 99.95*
SOURCER™ with BIOS Pre-Processor	$139.95*

(*OUTSIDE USA, ADD $15 SHIPPING; CA RESIDENTS, ADD SALES TAX)

All our products come with a 30 day money back satisfaction guarantee. Not copy protected. To order or receive additional information just call!

800-538-8157 x 811 (outside Calif.)
800-672-3470 x 811 (inside Calif.)

V COMMUNICATIONS

3031 Tisch Way, Suite 200, Dept. J1
San Jose, CA 95128
(408) 296-4224

PS/2, AT, XT and PC are trademarks of IBM Corp.

Number **125** on the Reader Service Card

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

```
        mov     di,cx   ;set aside the column
        and     cl,0111b ;keep only the column in-byte address
        shr     di,1
        shr     di,1
        shr     di,1    ;divide column by 8 to make a byte address
        add     di,ax   ;and point to byte
;
; Set up the GC rotation. In write mode 3, this is the rotation
; of CPU data before it is ANDed with the Bit Mask register to
; form the bit mask. Force the ALU function to "move". Uses the
; readability of VGA registers to leave reserved bits unchanged.
;
        mov     dx,GC_INDEX
        mov     al,GC_ROTATE
        out     dx,al
        inc     dx
        in      al,dx
        and     al,0e0h
        or      al,cl
        out     dx,al
;
; Set up BH as bit mask for left half, BL as rotation for right half.
;
        mov     bx,0ffffh
        shr     bh,cl
        neg     cl
        add     cl,8
        shl     bl,cl
;
; Draw all characters, left portion first, then right portion in the
; succeeding byte, using the data rotation to position the character
; across the byte boundary and then using write mode 3 to combine the
; character data with the bit mask to allow the set/reset value (the
; character color) through only for the proper portion (where the
; font bits for the character are 1) of the character for each byte.
; Wherever the font bits for the character are 0, the background
; color is preserved.
; Does not check for case where character is byte-aligned and
; no rotation and only one write is required.
;
; Draw the left portion of each character in the string.
;
        pop     cx      ;get back screen width
        push    si
        push    di
        push    bx
;
; Set the bit mask for the left half of the character.
;
        mov     dx,GC_INDEX
        mov     al,GC_BIT_MASK
        mov     ah,bh
        out     dx,ax
LeftHalfLoop:
        lodsb
        and     al,al
        jz      LeftHalfLoopDone
        call    CharacterUp
        inc     di      ;point to next character location
        jmp     LeftHalfLoop
LeftHalfLoopDone:
        pop     bx
        pop     di
        pop     si
;
; Draw the right portion of each character in the string.
;
        inc     di      ;right portion of each character is across
                        ; byte boundary
;
; Set the bit mask for the right half of the character.
;
        mov     dx,GC_INDEX
        mov     al,GC_BIT_MASK
        mov     ah,bl
        out     dx,ax
RightHalfLoop:
        lodsb
```



```
        and     al,al
        jz      RightHalfLoopDone
        call    CharacterUp
        inc     di      ;point to next character location
        jmp     RightHalfLoop
RightHalfLoopDone:
;
        pop     ds
        pop     bp
        pop     di
        pop     si
        pop     dx
        pop     cx
        pop     bx
        pop     ax
        ret
DrawString      endp
;
; Draw a character.
;
; Input:
;  AL = character
;  CX = screen width
;  ES:DI = address to draw character at
;
CharacterUp     proc    near
        push    cx
        push    si
        push    di
        push    ds
;
; Set DS:SI to point to font and ES to point to display memory.
;
        lds     si,[FontPointer]        ;point to font
;
; Calculate font address of character.
;
        mov     bl,14   ;14 bytes per character
        mul     bl
        add     si,ax   ;offset in font segment of character

        mov     bp,FONT_CHARACTER_SIZE
        dec     cx      ; -1 because one byte per char
CharacterLoop:
        lodsb                   ;get character byte
        mov     ah,es:[di]      ;load latches
        stosb                   ;write character byte
;
; Point to next line of character in display memory.
;
        add     di,cx
;
        dec     bp
        jnz     CharacterLoop
;
        pop     ds
        pop     di
        pop     si
        pop     cx
        ret
CharacterUp     endp
;
; Set the pointer to the font to draw from to ES:BP.
;
SelectFont      proc    near
        mov     word ptr [FontPointer],bp       ;save pointer
        mov     word ptr [FontPointer+2],es
        ret
SelectFont      endp
;
cseg    ends
        end     start
```

◆

PRODUCED 2005 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

DISK FORMAT CONVERSION

XenoCopy-PC

PC-DOS program lets your PC

$79.95 + $5.00 S/H
Sales Tax if CA.
VISA MasterCard UPS COD

READ / WRITE /
FORMAT / DUPLICATE
Disks from over 300 other micros

Upgrades available from all previous versions
for only $25.00 Call for Authorization

To Order Contact:

XENOSOFT™

1454 Sixth Street, Berkeley, CA 94710
(415) 525-3113

this was on purpose.

good software is it's own reward

Number **123** on the Reader Service Card

WINDOWS – MENUS – DATA ENTRY – SCREENS

HI-SCREEN XL™

If you program, you must try HI-SCREEN *XL* !

HI-SCREEN *XL* is the only tool that gives you all of the following (and much more) whatever programming language you use:

- Unlimited number of screens & windows
- Overlapping windows for menus, data entry, help screens
- Pull-down, pop-up and Lotus-style menus
- On-line help management
- Field checking (for type, range, format...)
- Field by field and/or full screen input modes
- Scrolling capability
- Full-featured WYSIWYG screen editor
- Redefinable function keys
- Data entry test mode under the editor
- True ease of use, and ...
- Ease of maintenance: modify screens without recompiling your program.

Includes the HI-SCREEN *XL* Toolbox:
- Capture existing screens from other applications
- Trace compiled programs
- *Print screens with clear data field description*
- Use screens/windows/menus from DOS and Batch files
- Create titles & logos with predesigned font screens
- ...

Softway, Inc.
PC/Soft Product Line
500 Sutter St., Suite 222
San Francisco, CA 94102

$149 - Risk Free
Visa, M/C welcome
Call (800) 338-2852
in CA (415) 397-4666

For PASCAL, C, dBASE, BASIC, COBOL, FORTRAN

Number **110** on the Reader Service Card

Software Tools

For Programmers & Non-Programmers

Get 'State of the Art' performance and save valuable time with these high quality utilities!

Opt-Tech Sort™

Opt-Tech Sort is a high performance Sort/Merge/Select utility. It can read, sort and write a file faster than most programs can even read the data. Example: 1,000 records of 80 bytes can be read, sorted and a new file written in less than 10 seconds (IBM XT). Opt-Tech Sort can be used as a stand-alone program or called as a subroutine to over 25 different programming languages.

All the sorting, record selection and reformatting facilities you need are included. A partial list of features includes: The ability to process files of any size. Numerous filetypes are supported including Sequential, Random, Delimited, Btrieve, dBASE II & III and many others. Up to 10 key fields can be specified (ascending or descending order). Over 16 different types of data supported. Powerful record selection capability allows you to specify which records are to be included on your output. Record reformatting allows you to change the structure of your output record and to output special fields such as record numbers for use as indexes.

MS-DOS $149. ★ NEW ★ Xenix $249.

★ NEW ★ VERSION On-Line Help™ ★ NEW ★ VERSION

On-Line Help allows you to easily add "Help Windows" to all your programs. On-Line Help is actually two help packages in one. You get BOTH Resident (pop-up) and Callable Help Systems.

The resident version allows you to add help to any system. Your Help System is activated when the "Hot Keys" that you specify are pressed. You can then chain between help windows in any manner you desire.

The callable version allows you to easily display help windows from your programs. A simple call to the help system makes the window appear. The original screen is automatically restored when the help window is cleared. On-Line Help is callable from over 20 different languages.

You have full control over the help window content, size, color and location.

MS-DOS $149. **Demo $10.** (apply toward purchase).

Scroll & Recall™

Scroll & Recall is a resident screen and keyboard enhancement. It allows you to conveniently scroll back through data that has gone off the top of your display screen. Up to 27 screens of data can be recalled or written to a disk file (great for documenting systems operations). Also allows you to easily recall and edit your previously entered DOS commands without retyping.

Scroll & Recall is very easy to use. It's a resident utility that's always there when you need it. MS-DOS $69.

Visa, M/C, AMEX, Check, Money Order, COD or Purchase Orders accepted.

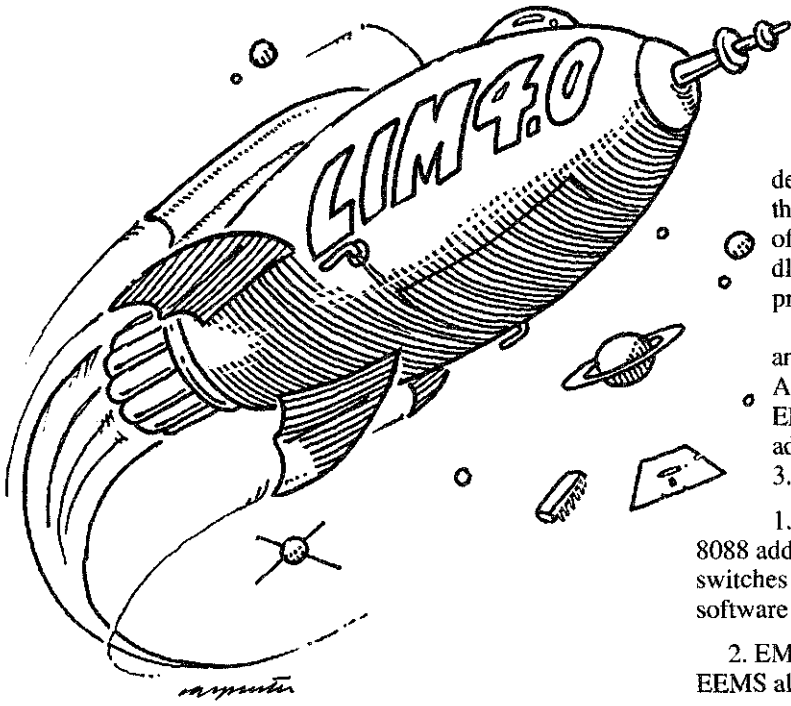
To order or to receive additional information just call and receive immediate highly qualified attention!

Opt-Tech Data Processing
P.O. Box 678 — Zephyr Cove, NV 89448
(702) 588-3737

Number **16** on the Reader Service Card

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

The LIM 4.0 ExpandedMemory Specification



by
John J. Newlin

When the first IBM PC was unveiled, industry observers marvelled at the 1 megabyte of memory addressable by the 8088 processor and the 640 kilobyte space manageable by DOS. But just as nature inevitably fills a vacuum, the 640 Kb of user space was filled with larger, more powerful applications programs. DOS itself grew in size and complexity, and in the short course of a few years, that seemingly mammoth 640 Kb of user space became very crowded. Large spreadsheets, TSRs, CAD/CAM software and the like were really straining the limits of the PC memory.

One solution to the problem of providing more memory was the introduction of the Intel 286, which can address up to 16 Mb of memory, and eventually the 386, capable of addressing up to 4 gigabytes. But the new CPU technology didn't provide relief for the millions of stock PC/XTs and clones in operation.

Intel Corporation, in concert with Lotus Development Corporation, devised a solution for the standard 8086/8088 based machine. In the spring of 1985, Intel and Lotus announced the availability of the Expanded Memory Specification (EMS) Version 3.0, which was designed to prevent the obsolescence of older 640 Kb limited machines. Microsoft Corporation subsequently joined Intel and Lotus, and Version 3.2 was released as the Lotus/Intel/Microsoft (LIM) expanded memory specification.

Expanded memory, as functionally defined by the LIM specification, embodies a bank-switched memory subsystem that embodies hardware expansion boards managed by a resident device driver that adheres to the specification. Although the EMS supported the addition of up to 8 Mb of expanded memory (2 Mb per board), properly handling more than one board involved some driver problems.

The next EMS development step was taken by another corporate trio, AST Research, Quadram, and Ashton-Tate (AQA), and was called the Enhanced EMS, or EEMS. EEMS was a superset of EMS and addressed some of the deficiencies inherent in EMS 3.2. The three major differences between the two are

1. With EMS, the mapping area (residing in the 8088 addressable space) is set physically with board switches or pre-defined at boot time. EEMS supports software definition of the mapping area at runtime.

2. EMS requires one contiguous 64 Kb mapping space. EEMS allows several non-adjacent areas to be defined.

3. EMS limits the mapping area to a maximum of 64 Kb. EEMS supports an area size of 16 Kb up to 1 Mb.

The introduction of EEMS was a significant step even though observers worried that a rivalry might develop between the two triumvirates that would cause standardization problems for the PC user base. EMS software would run on a board designed for the EEMS, but the converse was not necessarily true. Altogether, the EMS enhancements contained in the AQA EEMS made it possible to execute multiple programs in expanded memory. This process is called multitasking, and it became viable because the EEMS supports context-switching at processor speeds. Quarterdeck Office Systems was one of the first to take advantage of the inherent power of EEMS with their multitasking environment software called DESQview.

Even though EMS and EEMS provided up to 8 Mb of additional memory for the PC, there were still driver and board limitations that prevented applications software from exploiting the full potential of EMS technology. In August of this year Lotus, Intel, and Microsoft announced the release of LIM 4.0, a significant upgrade to the EMS specification designed to bolster the capabilities of MS-DOS by supporting much larger applications and multitasking, even on 8086/88 CPU machines. But the new LIM 4.0 specification is more than a major upgrade; it also clearly demonstrates that the industry's largest and most successful corporations are working in unison to prolong the useful life of DOS.

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

LIM 4.0 provides software developers with a host of new EMS functions that will make expanded memory a very attractive milieu in which to operate their products. Developers will find it much easier to write applications that utilize the power of expanded memory, many of which will contain functions available in programs designed to run in the OS/2 environment. Perhaps even more immediately significant, AST, Quadram, and Ashton-Tate have announced support for the new specification. LIM 4.0 includes the enhanced capabilities of the AQA EEMS specification, thus abrogating the need for EEMS as a separate specification and ending the confusion created by the existence of two EMS specifications.

What does LIM 4.0 provide that LIM 3.2 did not? The following is a summary of enhancements contained in LIM 4.0:

- increases the maximum expanded memory available from 8 Mb to 32 Mb
- doubles the number of functions and subfunctions available to developers
- provides support for execution of program code in expanded memory
- directly supports TSR programs in expanded memory
- permits cooperating programs to access and share common data
- supports multitasking

LIM 4.0 requires that handle values be in the range of 1 to 255 and that the upper byte of the returned word will always be zero and unusable by the application.

The LIM 4.0 specification consists of 30 numbered functions, many of which are comprised of subfunctions. The LIM 3.0 specification defined 15 functions. With two notable exceptions, the first 15 functions in LIM 4.0 are the same as those defined in the previous version. These exceptions are Function 4, Allocate Pages, and Function 5, Map/Unmap Handle Pages. Under LIM 3.0, an EMS driver could use any desired scheme for handle numbering. Some I've seen start with 0 and allocate handles consecutively from there while others start with a high word value and decrement as pages are allocated. LIM 4.0 requires that handle values be in the range of 1 to 255 and that the upper byte of the returned word will always be zero and unusable by the application. The value of zero is reserved for the operating system. A new feature has been added to Function 5 that permits the application to unmap physical pages, rendering them inaccessible for reading or writing.

Listing 1 is a table that contains a summary listing of the 30 LIM 4.0 functions by function number, name, and AH or AX register value setting for invocation. For each LIM function, there is also a Turbo Pascal procedure/function name that is a cross reference into Listing 2. Listing 2 contains a complete set of Turbo Pascal functions and procedures that implement the LIM 4.0 function calls. Several of the LIM 4.0 functions are reserved for operating system use only and are so labeled.

Note that in Listing 2, I have not provided any error handling code in the examples. This is because each programmer will probably want to set up his/her error handling somewhat differently and EMS function error handling will depend to some degree on the nature of the application. Note also that since at this writing I do not have a LIM 4.0 EMS driver, this code has not been tested in executable form.

For the remainder of this article I shall concentrate entirely upon the new functions LIM 4.0 adds to the previous version of the specification. Along the way I will provide some specific coding examples to illustrate important functionality and to demonstrate how some of the code in Listing 2 can be used.

Many of the new EMS functions involve providing access to an array of data, the size of which is indeterminate at runtime; for example, Function 15 (Get/Save Page Map) fills an array with the state of all mapping registers. A Subfunction 15 returns the number of bytes that will be transferred to the designated storage area. The following code illustrates how Turbo Pascal can be used to invoke Function 15:

```
var
   dest_ptr = ^byte;
   no_of_bytes : integer;

begin
   no_of_bytes := PageMapSize;
   getmem(dest_ptr,no_of_bytes);
   GetPageMap(dest_ptr^);
```

Note that the use of pointers and Turbo Pascal's intrinsic untyped variable parameter allows the programmer maximum flexibility in dealing with data structures returned by the EMS driver. Another, more complex, example is Function 21, which returns an array that contains all active handles and the names associated with the handles. The structure of this array is defined below along with some other useful types:

```
TYPE
   name_type = array[1..8] of char;
   handle_dir_type = record
                        handle_value : integer;
                        handle_name  : name_type;
                     end;
handle_array_type = array[1..5] of handle_dir_type;
handle_array_ptr = ^handle_array_type;
```

Now let's create a pointer to this structure and a variable to receive the number of active handles:

```
VAR
   handle_ptr : handle_array_ptr;
   handles : integer;
   name : name type;
```

First we have to call Function 12 to determine the number of active handles:

```
   handles := HandleCount;
```

Next, allocate the memory from the heap:

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

```
getmem(handle_ptr,handles * sizeof(handle_dir_type));
```

Then call the EMS driver to get a copy of the array into the heap:

```
handles := GetHandleDir(handle_ptr^);
```

To access a specific member of the array, simply use the following statement:

```
name := index_ptr^[4].handle_name;
```

Declaration of the size of the array is not important since no space is allocated to it at compile time. It is important, however, to ensure that compiler index range checking is turned off ({$R-}), which is the default.

As you've probably guessed by now, the LIM 4.0 specification supports (via Function 20) the assignment of eight-character handle names and the characters comprising the name can be of any ordinal value from 0 to 255. When a handle name is assigned, at least one of the characters must be non-zero because a handle name containing all zero characters has no name. Handle names can be used so that a family of applications can share common data in expanded memory—for example, a data base manager, word processor, and spreadsheet might share data in a way that allows the spreadsheet to access data left in expanded memory by the data base manager, etc. Function 21 (Search for Named Han-dle) can be invoked to obtain the EMM handle associated with a handle name, and Function 14 (Get Handle Pages) provides the number of expanded memory pages allocated to the handle.

Two additional functions that resident programs, interrupt service routines, and device drivers can use to save and restore the current mapping hardware state are included in LIM 4.0. Version 3.2 Functions 8 and 9 save and restore only the 64 Kb page frame while new Functions 15 and 16 allow saving the mapping state for the memory region external to the LIM 3.2 page frame. The difference between Functions 15 and 16 is that Function 15 saves the entire mapping state while Function 16 saves only a partial state. Unless it is necessary to save the entire mapping state, the programmer is advised to use Function 16 because it requires less memory and is potentially faster than Function 15.

LIM 4.0 also provides more flexible and powerful page mapping and unmapping functions. Function 5 is a carryover from LIM 3.2 and allows mapping of just one logical page to a physical page at a time. Function 17 provides the capability to map any number of logical pages into physical pages. Additionally, this function increases page mapping flexibility by providing the programmer with a choice of mapping logical pages to physical pages by specifying either the physical page number or the physical page segment address. This is important because LIM 4.0, unlike its predecessor, permits expanded memory to exist in non-contiguous blocks of the PC addressable memory space.

Function 17 is supported by Function 25, which will return an array containing the segment address and physical page number for each mappable physical page in the system. The programmer is thus provided with a cross reference between the available physical page numbers and the actual segment address where each page resides. A subfunction call will return a value equal to the number of mappable pages in the system so that the program may know how much space to reserve for the returned cross reference array.

DEVELOPMENT TOOLS FOR C

- A COLLECTION OF "TOOL BOXES" FOR THE DEVELOPMENT OF POWERFUL C PROGRAMS
- DESIGNED TO BE USED BY BOTH NEW AND EXPERIENCED C PROGRAMMERS
- EACH TOOL BOX IS FULLY DOCUMENTED -- WITH EXAMPLES

SYSTEM TOOL BOX
- ACCESS THE POWER OF YOUR PC
- KEYBOARD & SCREEN MANAGEMENT
- GRAPHICS & MUCH MORE
- 184 PAGE USER'S GUIDE

LIB + SRC $35

UTILITY TOOL BOX
- GENERAL PURPOSE FILTERS
- UTILITY PROGRAMS
- GENERAL PURPOSE FUNCTIONS
- 86 PAGE USER'S GUIDE

LIB + SRC $25

WINDOW TOOL BOX
- COMPLETE WINDOW CONTROL
- BUILD WINDOW DATABASES
- FULL ERROR REPORTING
- 84 PAGE USER'S GUIDE

LIB $45 -- SRC $125

MENU TOOL BOX
- VERTICAL/HORIZONTAL MENUS
- CREATE MENU DATABASES
- FULL ERROR REPORTING
- 78 PAGE USER'S GUIDE

LIB $35 -- SRC $75

- HARDWARE -- IBM PC, TI PRO OR TANDY 1000
- COMPILER -- MICROSOFT™ C (3.00 & LATER)
- CALL OR WRITE FOR FREE INFO & DEMO PROGRAM
- TECHNICAL SUPPORT IS UNLIMITED & FREE
- NO ROYALTIES

FLEMING SOFTWARE
5741 Wagonwheel Trail
Castle Rock, CO 80104
(303) 688 - 3584

Number 118 on the Reader Service Card

SRMS™

"...is the easiest to use of those [Revision Control Systems] with screen based menus..." PC Tech Journal Sept '87

"...is a practical and versatile program that many programmers will want to use daily" IEEE Software July '87

SRMS is a full featured revision control system that allows programmers, system managers, and librarians to get control of software projects. SRMS stores every version of a source program in a single ASCII file, with the ability to restore each and every version easily and quickly. Only changes (deltas) between versions are actually stored, so disk usage is drastically reduced. Each version is stored with a unique Version Identification String and unlimited user supplied comments, making identification and documentation tasks easy. SRMS supports projects written in any language, and allows you to use your current editor and compiler without conflict. Version MERGE facilities help consolidate parallel development efforts. SRMS also provides the ability to bind various versions of different modules together for reconstructing a particular software system release. DOS directory pathnames and environment variable support also aid you in your work. Full typeset documentation with tutorial and project management chapters.

Version 3.1 Now Available
- Up to 50% faster
- Enhanced Version Numbering
- Configuration File Support
- Comment Line Editing
- Library Module Extension Control
- Increased Wildcard Handling

Multi-User SRMS Now Available
- Same Features of Single User SRMS
- Journaling and Authorship Facilities
- Library Access Control Lists
- LAN Compatible

SRMS Integrates Fully with QMAKE

QMAKE is a powerful and flexible utility patterned after UNIX make, that can be used to control the rebuilding of simple or complex software systems. QMAKE is able to use the date/time stamps of source stored with SRMS to rebuild your systems quickly and accurately.

Text Manipulation Tools

TXTTOOLS is a set of three utilities that allow you to efficiently search for, edit and show the differences associated with text files.

Single User SRMS	$185
Multi-User SRMS	$250
SRMS 5 Site LAN	$750
QMAKE	$99
SRMS+QMAKE	$250
TXTTools	$85

30 Day Money Back Guarantee

QUILT COMPUTING
7048 Stratford Road
Woodbury, MN 55125
(612) 739-4650

Volume Discounts and Dealer Inquiries Welcome

Number 46 on the Reader Service Card

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

Loading and executing code in expanded memory has been greatly facilitated by the addition of Functions 22 (Alter Page Map and Jump) and 23 (Alter Page Map and Call Subfunctions). These two powerful functions permit the programmer to alter the expanded memory mapping and transfer program control to a specific point within expanded memory. As an example, consider the following scenario. You want to produce a program that requires over 150 Kb of memory for executable code, yet you anticipate machine environments where less than 40 Kb of conventional memory is likely. Using LIM 4.0, it is possible to write a main code module of less than 40 Kb and to place other required code in overlay type modules that will be loaded into expanded memory.

The ability to move or exchange areas of memory is another powerful addition to the LIM specification. Function 24 allows the programmer to move and exchange up to one megabyte of data with just a single function call. Data may be moved by using other functions, but the beauty of Function 24 is that it automatically checks for overlapping memory regions and performs all the necessary mapping, preserving the mapping context through the exchange/move call. The following memory region source/destination combinations are supported by the call:

- conventional memory to conventional memory
- conventional memory to expanded memory
- expanded memory to expanded memory
- expanded memory to conventional memory

Another new capability in LIM 4.0 is Function 18 (Reallocate Pages). The only way an executing program could real-

$99 BUSINESS BBS

24 hour business information center via modem (300-2400 baud). Setup custom multi-level menus, data entry forms and info-sheets easily with no programming req'd. Integrated data base, XMODEM up/downloads, remote PC operation. With source code $249.

$99 PC COMMX $119 CP/M

Emulates: VT100/102, Wyse, HP, ADM, TV, IBM, ADDS
Transfers: KERMIT, XMODEM, COMMX mainframe, TLX/TWX. POPUP hotkey to DOS or programs. Unattended control scripts, dial directory for 700 entries & electronic mail features.

$59 C DATA ENCRYPTION

Data Encryption Standard (U.S. government standard FIPS PUB46) in Microsoft "C". Includes compression & telecomm formatting, allowing faster transmission & less storage space plus compatibility with any computer or service. Complete "C" source code $249.

HAWKEYE GRAFIX Inc
BOX 1400, OLDSMAR FLORIDA 33557
DIAL 813-855-5846

Number 57 on the Reader Service Card

Introducing

Quaid Analyzer

the tool that created CopyWrite

Now you can debug your own programs with a professional quality debugger - the one that unraveled every form of copy-protection used on the PC.

With the Quaid Analyzer, you can:

- ☐ See occurences of any interupt, with its meaning shown on the screen.
- ☐ View memory as text or instructions, scrolling as easily as you do with an editor.
- ☐ Run until a memory location or I/O port is changed.
- ☐ Protect your hard disk from accidental destruction.
- ☐ Analyze software without the source, even when it uses countermeasures to thwart tracing.
- ☐ See all stages of the boot load.

We kept the Quaid Analyzer off the market to avoid helping publishers with copy-protection. Now that copy-protection is gone, we can sell it to you.

The Quaid Analyzer is a software tool occupying 100K bytes. It runs on any IBM PC and most MS-DOS systems without hardware modification.

To order Call **(416) 961-8243**

Quaid Analyzer $99 U.S.

All orders shipped at our expense within a day. All major credit cards accepted.

or return coupon to:
45 Charles St. East
Third Floor, Dept. 603
Toronto, Ontario. M4Y 1S2

Payment method MC-Visa-Amex-Diners-Check
Card No. _____
Expiry Date _____
Name _____
Address _____
City/State _____
Phone No. _____
Signature _____

Quaid Software Limited

Ask about Disk Explorer the program that takes over where Quaid Analyzer leaves off.

Contact this Advertiser directly

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

locate expanded memory in LIM 3.2 was to deallocate all of its pages with Function 6 and then call Function 4 to allocate a new set of pages. Function 18 is much more flexible in that it supports the following four reallocation cases:

- deallocate all assigned memory pages but retain the active handle
- increase assigned memory by reallocating more pages than are currently assigned
- decrease assigned memory by reallocating fewer pages than are currently assigned
- reallocate the same number of pages as currently assigned (used to check the returned status to verify a previous reallocation request)

Before closing, I want to say a few words about the location of two important program data structures, the stack and the Disk Transfer Area (DTA). Both LIM 3.2 and LIM 4.0 specifically admonish the programmer not to place the stack in expanded memory. With the stack in expanded memory, an interrupt driven routine might take control, page the interrupted program's stack out of the page frame, and attempt to use the stack, thus clobbering some area in the new memory page. It can be argued that any well behaved TSR, ISR, or device driver will always switch to a local stack prior to manipulating expanded memory, but it would certainly be safer not to assume that all such programs are well behaved. The same is true with the placement of the DTA. Although LIM 4.0 does not specifically mention the DTA, the same reasoning applies.

Space limitations prevent me from discussing all the new functions contained in LIM 4.0 and from discussing the functions I have addressed in greater detail. The purpose of this overview is to whet programmers' appetites for writing expanded memory applications. I urge anyone interested in designing LIM 4.0 based software to acquire the specification from Lotus, Intel, or Microsoft. Expanded memory is going to play an ever increasing role in the future of PC based applications. Now is an excellent time to get on board. ♦

Lotus Development Corporation
55 Cambridge Parkway
Cambridge, MA 02142

Intel Corporation
5200 NE Elam Young Parkway
Hillsboro, OR 97124

Microsoft Corporation
16011 NE 36TH Way
Box 97017
Redmond, WA 98073

John J. Newlin is a retired Naval Aviator who currently works for Hughes Aircraft Co. as a systems engineer. He holds degrees in mathematics and computer science and has programmed extensively in assembly, Turbo Pascal, Marshal Pascal, and Modula 2. Mr. Newlin is the author of SCOUT, a memory resident disk/file/directory manager (reviewed by PC Magazine, *February 24, 1987), and ARCMASTER, a "point and shoot" style interface between the user and ARC and de-ARC programs. Mr. Newlin is a charter member of the Association of Shareware Professionals. Readers who wish to contact Mr. Newlin may write to him at the following address:*

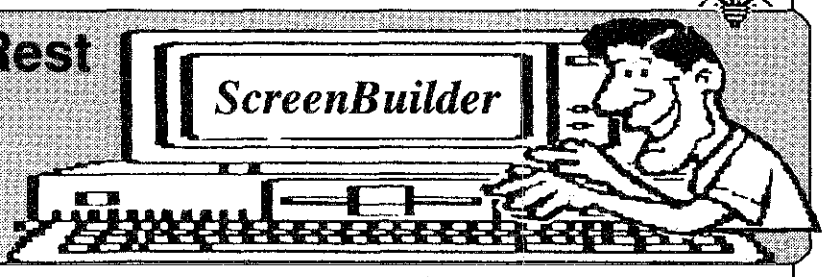
John J. Newlin
4060-228 Rosenda Court
San Diego, CA 92122

Code follows.

You've Seen all the Rest Now go for the Best!

Absolutely NO Royalties!

NOW SHIPPING Ver 2.10 with Data Entry Capabilities!



- **Create Interactive, Animated, Automated:**
  - Demos
  - Tutorials
  - User/Program Shells
  - Teaching Aids
  - Children's Education
  - Continuously Running Presentations
- **Orders of Magnitude above the rest:**
  - Can generate BASIC BLOADable Screens
  - Can generate Turbo C Source for Screens
  - Can generate ANSI files.
  - Sound and Music Capabilities
  - Mathematics and String Logic
  - User Inputs and Outputs
  - Printer Control
  - Loops, Subroutines, Decision Making
  - Screens are directly usable by Application Programs
  - Supports all 256 Colors and Characters
  - Screen Capture Program
  - Runs on CGA, EGA and Monochrome
  - Much, Much More!
- **Easy to use. Has context sensitive help.**
  - Complete, comprehensive 150+ page User Manual guides you through every step of the way.

Just $59.95 Complete!

Sample Demo Disk available for $2.00 (Credited towards your first order)

Call Today to order by credit card or send $59.95 plus $3.00 shipping/handling to:

GILMORE SYSTEMS
P. O. Box 3831
Beverly Hills, CA 90212-0831
(213) 275-8006

Requires IBM PC, XT, AT or compatible. DOS 2.0 or higher. 320 K RAM minimum. Fixed disk recommended.

Number **135** on the Reader Service Card

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

Listing 1. LIM 4.0 EMS SPECIFICATION FUNCTION SUMMARY

This table presents a sequential list of EMM 4.0 functions. Each entry consists of the function number, function name, hexidecimal value for the AH register, and Turbo Pascal function or procedure name. Subfunctions occupy multiple lines under the applicable function number, and each line contains the AX register (AH/AL pair) value for the function call.

Number	Function Name	AH/AX value	Turbo Pascal Example
1	Get Status	40H	function GetStatus
2	Get Page Frame Address	41H	function PageFrame
3	Get Unallocated Page Count	42H	function FreePages
4	Allocate Pages	43H	procedure GetPages
5	Map/Unmap Handle Pages	44H	procedure MapPages
6	Deallocate Pages	45H	procedure GivePages
7	Get Version	46H	function EmmVersion
8	Save Page Map	47H	procedure SavePageMap
9	Restore Page Map	48H	procedure RestPageMap
10	Reserved - Not for applications use		
11	Reserved - Not for applications use		
12	Get Handle Count	4BH	function HandleCount
13	Get Handle Pages	4CH	function HandlePages
14	Get All Handle Pages	4DH	function GetAllHandles
15	Get/Set Page Map Subfunctions		
	Get Page Map	4E00H	procedure GetPageMap
	Set Page Map	4E01H	procedure SetPageMap
	Get & Set Page Map	4E02H	procedure GetSetMap
	Get Size of Page Map	4E03H	function PageMapSize
16	Get/Set Partial Page Map Subfunctions		
	Get Partial Map	4F00H	procedure GetPartMap
	Set Partial Map	4F01H	procedure SetPartMap
	Get Size of Partial Page Map	4F02H	procedure PartMapSize
17	Map/Unmap Multiple Handle Pages Subfunctions		
	Logical Page/Physcial Page	5000H	procedure MapLogToPhys
	Logical Page/Segment Address	5001H	procedure MapLogToSeg
18	Reallocate Pages	51H	function ReAllocPages
19	Get/Set Handle Attribute Subfunctions		
	Get Handle Attribute	5200H	function GetHandleAttr
	Set Handle Attribute	5201H	procedure SetHandleAttr
	Get Attribute Capability	5202H	function GetAttr
20	Get/Set Handle Name Subfunctions		
	Get Handle Name	5300H	procedure GetHandleName
	Set Handle Name	5301H	procedure SetHandleName
21	Get Handle Directory Subfunctions		
	Get Handle Directory	5400H	function GetHandleDir
	Search for Named Handle	5401H	function GetHandleValue
	Get Total Handles	5402H	function GetTotalHandles
22	Alter Page Map and Jump	55H	procedure EmmJump
23	Alter Page Map and Call Subfunctions		
	Alter Page Map and Call	560nH	procedure EmmCall
	Get Page Map Stack Space Size	5602H	function MapStackSize
24	Move/Exchange Memory Region Subfunctions		
	Move Memory Region	5700H	procedure MovMemRegion
	Exchange Memory Region	5701H	procedure ExcMemRegion
25	Get Mappable Physical Address Array Subfunctions		
	Get Mappable Physical Address	5800H	procedure PhysMapArray
	Get Mappable Physical Entries	5801H	function PhysMapEntries
26	For use by operating systems only		
27	For use by operating systems only		
28	For use by operating systems only		
29	Prepare EM for Warm Boot	5CH	procedure PrepWarmBoot
30	For use by operating systems only		

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

Code from "LIM 4.0 EMS" by John Newlin

1

Listing 2

```
(*    TURBO PASCAL LIM 4.0 EXPANDED MEMORY CODING EXAMPLES     *)
(*-------------------------------------------------------------*)
(* Most of the TYPE declarations in this listing are provided  *)
(* as Pascal templates for data structures defined in the LIM  *)
(* 4.0 EMS specification. Most of the function/procedure VAR   *)
(* parameters are declared as Turbo Pascal untyped             *)
(* parameters due to the fact that the size of the data        *)
(* structures will have to be determined dynamically at the    *)
(* time of the function/procedure call. Note that this code    *)
(* is not a program, unit, or other compilable structure and   *)
(* has not been tested in executable form.                     *)

TYPE

  Str4 = string[4];

  PartMapPtrType = ^integer;

  HandleNameType = array[1..8] of char;

  Regpack     = record case integer of
                  1 : (ax,bx,cx,dx,bp,si,di,ds,es,flags:integer);
                  2 : (al,ah,bl,bh,cl,ch,dl,dh : byte);
                end;

  HandlePageRec = record
                    emm_handle,pages : integer;
                  end;

  MapArrayType = array[0..255] of byte;

  TargetAddrPtrType = ^integer;

  MapArrayPtrType = ^integer;

  MapJumpType = record
                  target_addr_ptr : TargetAddrPtrType;
                  map_length      : byte;
                  map_array_ptr   : MapArrayPtrType;
                end;

  MapCallType = record
                  target_addr_ptr   : TargetAddrPtrType;
                  new_map_length    : byte;
                  new_map_array_ptr : MapArrayPtrtype;
                  old_map_length    : byte;
                  old_map_array_ptr : MapArrayPtrType;
                  reserved          : array[1..4] of integer;
                end;

  LongIntType = Array[1..2] of integer;

  SourceDestType = record
                     region_length      : LongIntType;
                     source_mem_type    : byte;
                     source_handle      : integer;
                     source_init_offset : integer;
                     source_init_seg    : integer;
                     dest_mem_type      : byte;
                     dest_handle        : integer;
                     dest_init_offset   : integer;
                     dest_init_seg      : integer;
                   end;
```

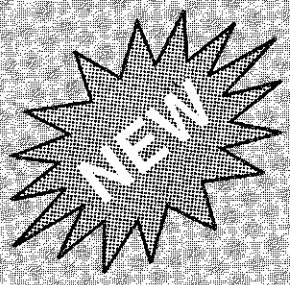
2

```
CONST

 Emm = $67;

VAR
  Regs : Regpack;
  HandlePageArray : array[1..255] of HandlePageRec;

Function GetStatus : integer;
begin
  Regs.ah := $40;
  intr(Emm,Regs);
  GetStatus := Regs.ah
end;

Function PageFrame : integer;
begin
  Regs.ah := $41;
  intr(Emm,Regs);
  PageFrame := Regs.bx
end;

Function FreePages(var TotalPages : integer) : integer;
begin
  Regs.ah := $42;
  intr(Emm,Regs);
  TotalPages := Regs.dx;
  FreePages := Regs.bx;
end;

Procedure GetPages(var Handle,Pages : integer);
begin
  Regs.ah := $43;
  Regs.bx := Pages;
  intr(Emm,Regs);
  Handle := Regs.dx;
end;

Procedure MapPages(Handle,Logical,Physical : integer);
begin
  Regs.ah := $44;
  Regs.al := Physical;
  Regs.bx := Logical;
  Regs.dx := Handle;
  intr(Emm,Regs);
end;

Procedure GivePages(Handle : integer);
begin
  Regs.ah := $45;
  Regs.dx := Handle;
  intr(Emm,Regs);
end;

Function EmmVersion : str4;
var lobyte : byte;
    s1,s2 : str4;
begin
  Regs.ah := $46;
  intr(Emm,Regs);
  lobyte := Regs.al shl 4;
  lobyte := lobyte shr 4;
  str(Regs.al shr 4,s1);
  str(lobyte,s2);
  EmmVersion := s1 + '.' + s2;
end;
```

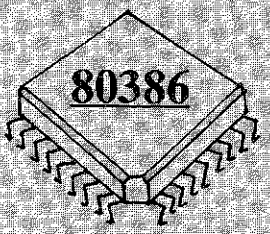
Code continues...

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

SERIOUS DEBUGGING AT A REASONABLE PRICE



Soft-ICE™



All the speed and power of a hardware-assisted debugger at a software price - $386

Features

Real-time break points on:
- Memory locations
- Memory ranges
- Execution
- I/O ports
- Interrupts (hardware and software)

Dual monitor support

System memory map

Regain control with a keystroke

Even with the following code:

```
CLI
MOV   AL,0FFH
OUT   21H,AL
JMP   $
```

and much, much more

How Soft-ICE works

Soft-ICE unleashes the power of the 80386 to surround your program in a virtual machine. This gives you complete control of the DOS environment. Soft-ICE uses 80386 protected mode features, such as paging, I/O privilege level, and break point registers, to add real-time hardware-level breakpoints to your existing DOS debugger. To use Soft-ICE you simply pop the Soft-ICE window up through a key sequence, set your hard break points, then return to your soft debugger. As the target program is executing, Soft-ICE recognizes when the breakpoint conditions have been reached and gives control back to your soft debugger. And this is all done at full 80386 speed! Soft-ICE can also be used in stand-alone mode. This comes in handy if you are debugging loadable device drivers, interrupt handlers, or terminate and stay resident programs. All of the standard debugging features are available to help you find the most difficult systems problems.

Benefits

- **Works with CodeView** --To get you up and going as fast as possible, Soft-ICE is designed to work with your existing software debuggers such as CodeView and Pericope I & II.
- **Breaks the 640K barrier** -- If you have extended memory, Soft-ICE takes up **ZERO** bytes of memory in the first 1MB of address space. This means you can load and debug your largest programs.
- **Power of an in-circuit emulator** -- At only $386, you can give every member of your software development team the power of an in-circuit emulator.
- **AT compatible 80386 PC's** -- Soft-ICE works with all AT compatible 80386 PC's, such as COMPAQ's Deskpro 386 and the IBM Model 80.
- **Easy to learn** -- Soft-ICE is so easy to learn that you can be finding bugs with Soft-ICE by the time you could install a hardware-assisted debugger.

"Since Soft-ICE doesn't take up any of my memory I have it in my AUTOEXEC.BAT to load every day... It has saved me at least one months time on my latest device driver program." Peter Ricker, President of Maverick Software

30 day money-back satisfaction guarantee. Visa and Master Card accepted. Ask about our coupon program.

To order or to get more information, call (603) 888-2386

NU-MEGA TECHNOLOGIES

P.O. BOX 7607
NASHUA, NH 03060-7607

Number **160** on the Reader Service Card

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

Newlin Code, Continued

3

```
Procedure SavePageMap(Handle : integer);
begin
  Regs.ah := $47;
  Regs.dx := Handle;
  intr(Emm,Regs);
end;

Procedure RestSaveMap(Handle : integer);
begin
  Regs.ah := $48;
  Regs.dx := Handle;
  intr(Emm,Regs);
end;

Function HandleCount : integer;
begin
  Regs.ah := $4B;
  intr(Emm,Regs);
  HandleCount := Regs.bx;
end;

Function HandlePages(Handle : integer) : integer;
begin
  Regs.ah := $4C;
  Regs.dx := Handle;
  intr(Emm,Regs);
  HandlePages := Regs.bx;
end;

Function GetAllHandles : integer;
begin
  Regs.ah := $4D;
  Regs.es := seg(HandlePageArray);
  Regs.di := ofs(HandlePageArray);
  intr(Emm,Regs);
  GetAllHandles := Regs.bx;
end;

Procedure GetPageMap(var maparray);
begin
  Regs.ax := $4E00;
  Regs.es := seg(maparray);
  Regs.di := ofs(maparray);
  intr(Emm,Regs);
end;

Procedure SetPageMap(var maparray);
begin
  Regs.ax := $4E01;
  Regs.ds := seg(maparray);
  Regs.si := ofs(maparray);
  intr(Emm,Regs);
end;

Procedure GetSetMap(var getarray,setarray);
begin
  Regs.ax := $4E02;
  Regs.es := seg(getarray);
  Regs.di := ofs(getarray);
  Regs.ds := seg(setarray);
  Regs.si := ofs(setarray);
  intr(Emm,Regs);
end;

Function PageMapSize : integer;
begin
  Regs.ax := $4E03;
```

4

```
  intr(Emm,Regs);
  PageMapSize := Regs.al;
end;

Procedure GetPartMap(var part_page_map,dest_array);
begin
  Regs.ax := $4F00;
  Regs.ds := seg(part_page_map);
  Regs.si := ofs(part_page_map);
  Regs.es := seg(dest_array);
  Regs.di := ofs(dest_array);
  intr(Emm,Regs);
end;

Procedure SetPartMap(var source_array);
begin
  Regs.ax := $4F01;
  Regs.ds := seg(source_array);
  Regs.si := ofs(source_array);
  intr(Emm,Regs);
end;

Function PartMapSize(pages : integer) : integer;
begin
  Regs.ax := $4F02;
  Regs.bx := pages;
  intr(Emm,Regs);
  PartMapSize := Regs.al;
end;

Procedure MapLogToPhys(var map_array; handle,map_array_len : integer);
begin
  Regs.ax := $5000;
  Regs.dx := handle;
  Regs.cx := map_array_len;
  Regs.ds := seg(map_array);
  Regs.si := ofs(map_array);
  intr(Emm,Regs);
end;

Procedure MapLogToSeg(var seg_array; handle,seg_array_len : integer);
begin
  Regs.ax := $5001;
  Regs.dx := handle;
  Regs.cx := seg_array_len;
  Regs.ds := seg(seg_array);
  Regs.si := ofs(seg_array);
  intr(Emm,Regs);
end;

Function ReAllocPages(handle,alloc_count : integer);
begin
  Regs.ah := $51;
  Regs.bx := alloc_count;
  Regs.dx := handle;
  intr(Emm,Regs);
  ReAllocPages := Regs.bx;
end;

Function GetHandleAttr(handle : integer) : integer;
begin
  Regs.ax := $5200;
  Regs.dx := handle;
  intr(Emm,Regs);
  GetHandleAttr := regs.al;
end;
```

Code continues...

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

Newlin Code, Continued

5

```
Procedure SetHandleAttr(handle,attr : integer);
begin
  Regs.ax := $5201;
  Regs.bl := attr;
  Regs.dx := handle;
  intr(Emm,Regs);
end;

Function GetAttr : integer;
begin
  Regs.ax := $5202;
  intr(Emm,Regs);
  GetAttr := Regs.al;
end;

Procedure GetHandleName(handle : integer; var name : HandleNameType);
begin
  Regs.ax := $5300;
  Regs.dx := handle;
  Regs.es := seg(name);
  Regs.di := ofs(name);
  intr(Emm,Regs);
end;

Procedure SetHandleName(handle : integer; var name : HandleNameType);
begin
  Regs.ax := $5301;
  Regs.dx := handle;
  Regs.ds := seg(name);
  Regs.si := ofs(name);
  intr(Emm,Regs);
end;

Function GetHandleDir(var handle_dir) : integer;
begin
  Regs.ax := $5400;
  Regs.es := seg(handle_dir);
  Regs.di := ofs(handle_dir);
  intr(Emm,Regs);
  GetHandleDir := Regs.al;
end;

Function GetHandleValue(var name : HandleNameType) : integer;
begin
  Regs.ax := $5401;
  Regs.ds := seg(name);
  Regs.si := ofs(name);
  intr(Emm,Regs);
  GetHandleValue := Regs.dx;
end;

Function GetTotalHandles : integer;
begin
  Regs.ax := $5402;
  intr(Emm,Regs);
  GetTotalHandles := Regs.bx;
end;

Procedure EmmJump(handle : integer; var map_jump_rec; page_selector : byte);
begin
  Regs.ah := $55;
  Regs.al := page_selector;
  Regs.dx := handle;
  Regs.ds := seg(map_jump_rec);
  Regs.si := ofs(map_jump_rec);
  intr(Emm,Regs);
end;
```

6

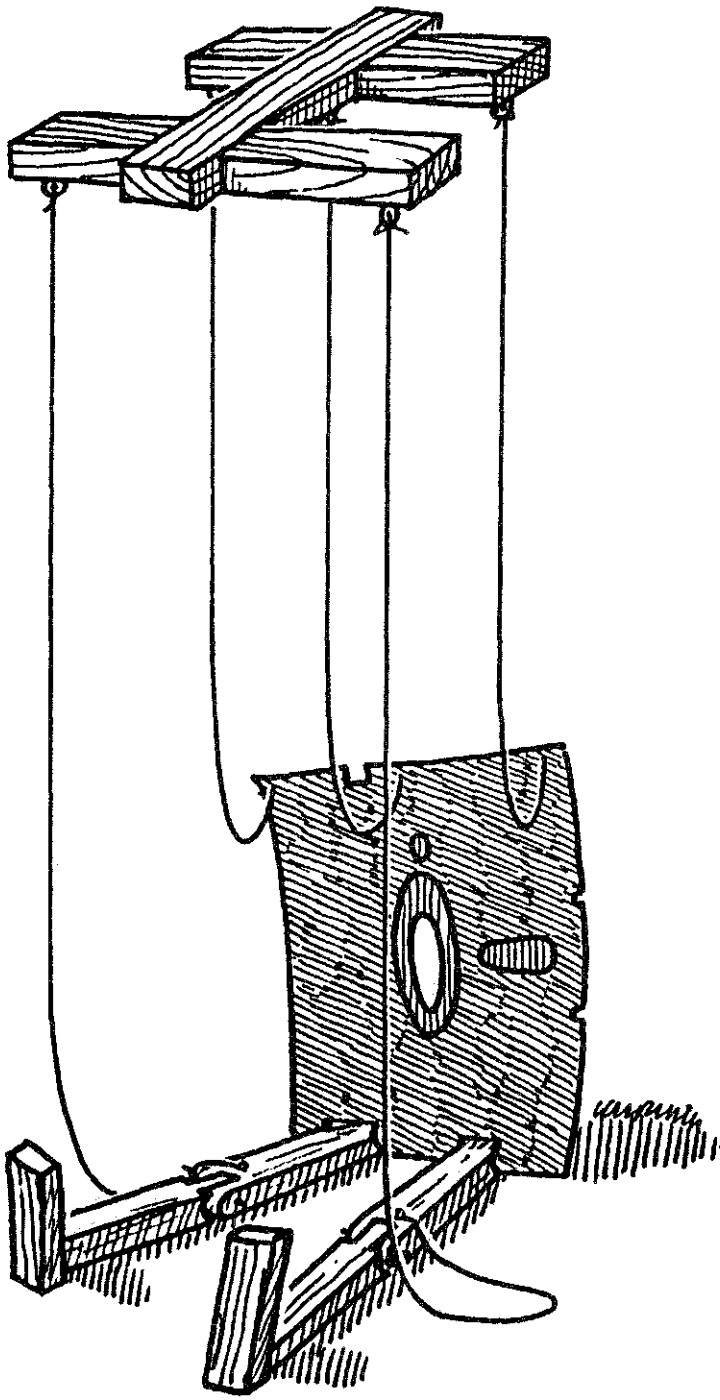
```
Procedure EmmCall(handle : integer; var map_call_rec; page_selector : byte);
begin
  Regs.ah := $56;
  Regs.al := page_selector;
  Regs.dx := handle;
  Regs.ds := seg(map_call_rec);
  Regs.si := seg(map_call_rec);
  intr(Emm,Regs);
end;

Function MapStackSize : integer;
begin
  Regs.Ax := $5602;
  intr(Emm,Regs);
  MapStackSize := Regs.bx;
end;

Procedure MovMemRegion(var source_dest_rec);
begin
  Regs.ax := $5700;
  Regs.ds := seg(source_dest_rec);
  Regs.si := ofs(source_dest_rec);
  intr(Emm,Regs);
end;

Procedure ExcMemRegion(var source_dest_rec);
begin
  Regs.ax := $5701;
  Regs.ds := seg(source_dest_rec);
  Regs.si := seg(source_dest_rec);
  intr(Emm,Regs);
end;

Procedure PhysMapArray(var phys_map_array; var map_size : integer);
begin
  Regs.ax := $5800;
  Regs.es := seg(phys_map_array);
  Regs.di := ofs(phys_map_array);
  intr(Emm,Regs);
  map_size := Regs.cx;
end;

Function PhysMapEntries : integer;
begin
  Regs.ax := $5801;
  intr(Emm,Regs);
  PhysMapEntries := Regs.cx;
end;

Procedure PrepWarmBoot;
begin
  Regs.ah := $5C;
  intr(Emm,Regs);
end;
```

♦

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

Discovering Your Environment

by
Tom Swan



Suppose you write a program that stores a temporary file on disk. You want people to tell the program where to create that file. One person might direct the file to a scratch diskette. Another programmer might decide to create the file in a miscellaneous subdirectory on a hard drive. The high-octane power user would no doubt prefer saving the file on a RAM disk for peak performance.

One way to give such information to programs is to save default path names and other variables in a setup file, which people can edit to direct output to specific drives and directories. This works but now requires the program to open and read the setup file in a known location—the very problem we are trying to solve.

A better solution, and one that many programmers neglect, is to use an environment variable to change the path where the program creates the file. For example, if you have a RAM disk D:, you could insert this command in your AUTOEXEC.BAT file:

```
SET TEMP=D:\
```

This creates the environment variable TEMP and gives it the setting D:\. When the program begins, if it finds TEMP in the environment, it knows to store the file on D:\—otherwise, the program uses a default path name. This avoids using a setup file to accomplish the same job.

The Environment Block

DOS saves all environment variables in a block of memory from 127 up to 32 Kbytes long. Each variable has the form,

```
<variable>=<setting><null>
```

The <variable> is an ASCII string, usually in uppercase. After the required equal sign comes the <setting>, also an ASCII string but possibly containing upper- and lowercase letters plus controls. A <null> character ends the construction.

The environment block may contain several such variables. It always contains at least one, COMSPEC, which tells DOS where to locate COMMAND.COM. Other typical environment variables are PATH, which specifies alternate batch and program file directory paths, and PROMPT, which changes the DOS command-line prompt.

To hold new variables or changes to existing variables, DOS automatically expands environment space as needed. The expansion can fail if you load terminate and stay resident

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

(TSR) programs. Doing this locks the environment block into an immovable island in memory, preventing DOS from expanding the block's size. If you receive an Out of Environment Space error when typing a SET command, a TSR is probably the cause.

The solution to the problem is unfortunately messy. If you execute all your SET commands before loading resident utilities, you solve the problem but might waste memory. Because DOS passes a copy of the environment block to new processes, each TSR receives its own copy of the entire environment. In other words, expanding the environment to 4K and starting five resident programs wastes 20K.

One answer is to load your TSRs and then start a second copy of COMMAND.COM. The TSRs receive copies of the default environment, usually only 127 bytes long. The secondary command processor also receives a copy of the environment block. Because this block is created after loading the resident programs, though, DOS is again able to expand the environment to hold new variables. To start a new command processor, make sure the current directory holds COMMAND.COM and type

```
COMMAND /P/E:4096
```

To avoid rerunning AUTOEXEC.BAT, copy COMMAND.COM to a different directory before typing this command. The /P tells DOS to make the secondary command processor permanent. The /E:4096 reserves 4096 bytes of memory for environment variables. According to the DOS manual, you can specify from 160 to 32768 bytes when expanding the environment block this way.

Turbo Pascal's PrefixSeg Word variable, defined in the System Unit, is new to Version 4.0. . . . Instead of using CSeg to locate PSP, use the new PrefixSeg variable.

One negative is the memory that the secondary command processor occupies. On my system, I count 3,472 bytes used after restarting COMMAND.COM. Obviously, you need to consider whether a secondary command processor is worth wasting this much memory. With a 4K environment and five TSRs, the net savings should be about 16–17K. With a small environment, the secondary processor might not be worth the trouble.

Reading the Environment

After using SET to create environment variables, Turbo Pascal 4.0 programs can search the environment block looking for various settings. Listing 1, ENV.PAS, demonstrates how to read environment string settings. When you compile and run **Env**, the program displays the settings for a few typical environment variables. Figure 1 shows what **Env** displays on my system.

```
Test reading environment strings


COMSPEC....C:\COMMAND.COM
PATH.......F:;C:\UTIL\BATS;C:\UTIL\TOOLS;C:\UTIL\DOS
PROMPT.....$p$g
TEMP......
```

Figure 1. Example output from ENV.PAS

At this point, TEMP has no setting. To change this, type SET TEMP D:\ and rerun the program. Experiment with various PROMPT and PATH commands to see the effect they have on the environment variables of the same names.

To read environment strings in your own programs, copy the **ENV.PAS CharPtr TYPE** declaration plus routines **AdvanceCP**, **GetEnv**, **Match** (nested inside **GetEnv**), and **ASCIIZtoStr** to your program. The following describes how to use these routines.

AdvanceCP

This procedure adds one to the offset of a **CharPtr** pointer variable. The program calls **AdvanceCP** to advance a pointer to successive environment characters in memory.

You might be able to replace calls to **AdvanceCP** with **Inc(p)** statements. Early Turbo Pascal 4.0 documentation indicates that this should work although it doesn't in the prerelease compiler I'm using to write this article.

GetEnv

The **GetEnv** function searches the environment for a variable name. If it finds the name, it returns a **CharPtr** pointer to the first character of the variable's setting. If it does not find the name, **GetEnv** returns NIL.

To search for an environment string, declare a **CharPtr** variable (**p**) and call **GetEnv** this way:

```
VAR p : CharPtr;
. . .
p := GetEnv( 'TEMP' );
IF p <> NIL
   THEN Writeln( 'TEMP is defined' )
ELSE Writeln( 'TEMP is not defined' );
```

Nested inside **GetEnv** is function **Match**, which returns true if parameter string **s** exactly matches the environment variable that **CharPtr p** addresses. The environment variable must end in an equal sign for **Match** to return true.

Continued...

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

In the main body of **GetEnv**, the first statement initializes **p** to address the first character of the environment block. To find this address, the program uses Turbo Pascal's MemW array—a device for viewing memory as an array of 16-bit words. The segment address of the environment block is stored at offset 002Ch in the Program Segment Prefix (PSP). This address plus offset 0 points **p** to the environment block.

Turbo Pascal's PrefixSeg Word variable, defined in the System Unit, is new to Version 4.0. Under earlier Turbo Pascal versions, you could locate the environment segment address at CSeg:002C. This no longer works because compiled programs and units now go in their own memory segments. Instead of using CSeg to locate PSP, use the new PrefixSeg variable.

The rest of **GetEnv** should be easy to understand. A WHILE loop cycles while **p** does not address the null that ends the environment block. The IF statement checks for a match on string **s** at address **p**, ending **GetEnv** when finding the environment variable. An inner WHILE loop skips unmatched environment ASCIIZ strings.

ASCIIZtoStr

This procedure converts ASCIIZ strings to length-byte string variables. An ASCIIZ string is a sequence of characters ending in a null character.

Using **ASCIIZtoStr** along with **GetEnv** gives you a simple way to read environment variables. Suppose you want to read the TEMP variable, using the setting as the path name for a new file. First declare a string variable, **pathName**. (The String type in Turbo Pascal 4.0 defaults to 255 characters.) Also declare a file variable **(f)**. For this example, I'll use a plain file type although any other file will also work.

```
VAR pathName : String;
    f : FILE;
```

Next, search the environment for the current TEMP setting. Use either this setting or a default as the file's path.

```
ASCIIZtoStr( GetEnv( 'TEMP' ), pathName );
IF Length( pathName ) = 0
   THEN pathName := 'C:\';      { default }
Assign( f, pathName + 'TEMP.$$$' );
Rewrite( f );
```

The effect of this is to create file TEMP.$$$ on C:\ by default or on the path specified by the TEMP environment variable. Typing SET TEMP=D:\ before running a program redirects the temporary file without recompiling the program and without a setup file.

Environment variables are useful alternatives for giving programs information. They help avoid creating and reading setup files while making use of a DOS feature that's too often ignored. The ENV.PAS listing shows how to discover the environment in your own Turbo Pascal 4.0 programs, a technique that belongs in every programmer's toolkit. ♦

Tom Swan is the author of Mastering Turbo Pascal *and* Mastering Turbo Pascal Files *published by Howard W. Sams, Inc., and* Programming With Macintosh Turbo Pascal *published by John Wiley & Sons, Inc.*

Code follows.

Software Tools

❑ **The DATE UTILITIES** let you copy, erase, or display files based on their name **and** date. You can also change a file's time stamp. All utilities support wildcards for file names and let you limit the files chosen to those created **before, on,** or **after** a specified date. The package includes DCOPY, DERASE, DDIR, CDATE, and three other general purpose utilities. Version 1.5 costs $47.50.

❑ **PROINDEX** is a professional document indexing package. It works with all word processed text to create an index in any style. PROINDEX compares your text file to a **list** of keywords or phrases. These keywords **or** phrases can be up to 70 characters long! No limit on the size of the file to index or the number of references. Version 2.24 costs $89.95.

❑ **Soft Fonts** for the HP LaserJet and compatible printers. We offer **inexpensive** Helv, Roman, and Script font families. We have Roman fixed width, Bar code 39, Title fonts, Symbol sets, and Credit cards. All font families are $50 or less! Printer drivers for most word processors cost only $15 per font family! Custom fonts & a download utility are also available.

All tools work on the IBM PC or any compatible, MS-DOS 2.1 & up. No charge for shipping in the U.S. Thirty day money back guaranty!

ECI P.O. Box 61, Wasco, Illinois 60183

VISA 312-377-3520 MC

Number **70** on the Reader Service Card

TURBO SOURCE SEARCH

"BBS-by-mail"

2 FREE DISKS then just $2.50 A DISK

TURBO SOURCE SEARCH (TSS) is the largest SIG library of Turbo Pascal public domain/shareware programs.

TSS is in fact a Bulletin Board System (BBS). We have the usual modem-connect BBS (617-545-9131), but also offer a BBS-by-mail service where members may get the files from the library on disks by mail for just $2.50.

TSS members pay a one time fee of $25 and receive a set of catalog disks plus credit for 2 FREE DISKS.

TSS has over 110 disks, 35Mb of compressed files, of software. These files have been placed into public access channels by their authors and collected by TSS so that you can avoid the costs of collecting them yourself.

And since TSS is turning ONE YEAR OLD we are offering a one price special of $250 for full membership PLUS the full library of disks updated through disk 150 which should be added to the library by mid-summer!!! (Current members: $2 per disk, 50 + disk orders.)

For a set of catalog disks send $5, or send $25 to become a full member of TSS, the "BBS-by-mail" to:

TURBO SOURCE SEARCH
P.O. BOX 876, SCITUATE, MA 02066, 617-545-6677
data/BBS: 617-545-9131 (anyone may call!)

VISA, MASTERCARD, COD $5 extra, non-usa orders add $10

Number **35** on the Reader Service Card

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

Upgrade your technology

The software technology available to programmers of IBM-compatible personal computers is truly amazing. And newer, more powerful development packages appear all the time. But until now, finding out about these important products has been a difficult and time consuming task.

FREE Buyer's Guide. The New Programmer's Connection Buyers Guide contains individual descriptions of over 500 titles of programmer's development software by over 150 manufacturers. Each description covers major product features as well as any software or hardware requirements and version numbers. In the box on the right are some examples of the types of descriptions you'll find in our Buyer's Guide.

No Hidden Charges. The low discount prices in our Buyer's Guide are all you pay. We don't charge extra for domestic UPS Ground shipping, credit cards, COD orders, purchase orders, sales tax (except Ohio) or special handling (except for non-Canadian international orders).

Guarantees. We offer FREE 30-day no-risk return guarantees and 30-day evaluation periods on most of our products.

Latest Versions. The products we carry are the latest versions and come with the same manufacturer's technical support as if buying direct.

Large Inventory. We have one of the largest inventories of programmer's development products in the industry. Most orders are shipped within 24 hours.

Noncommissioned Staff. Our courteous salespeople are always ready to help you. And if you aren't sure about your needs, our knowledgeable technical people can give you sound, objective advice.

Experience. We've specialized in development software for IBM-compatible personal computers since 1984 and are experienced at providing a full range of quality products and customer services.

How to Get Your Copy. There are three ways for you to receive your FREE copy of the Programmer's Connection Buyer's Guide: 1) Use the reader service card provided by this journal; 2) Mail us a card or letter with your name and address; or 3) Call one of our convenient toll free telephone numbers.

If you haven't yet received your Programmer's Connection Buyer's Guide, act now. Upgrading your programming technology could be one of the wisest and most profitable decisions you'll ever make.

Limited Time Only!

Sale Prices on ISAM File Managers

through 1/31/88

FairCom

c-tree in C Source Code
List $395 Reg $315 **Sale $289**

c-tree with r-tree
List $650 Reg $519 **Sale $499**

These fast and highly portable B+Tree functions provide multi-key ISAM file management for C programs. There are low level functions for directly accessing data and index files and high level functions for creating and manipulating ISAM files. The highly portable C source code can be compiled with almost any C compiler or computer for single-user, multi-user or network applications. It supports: record locking for multi-users; fixed and variable length records; fixed and variable length keys with key compression; re-use of deleted record space; duplicate and unique key fields; and more. The package includes a complete family of setup and maintenance utilities, unlimited technical support, no royalties, and free hardcopy listings of release updates. r-tree is an optional report generation utility for c-tree that permits complex, multi-line reports to be produced from single or multiple c-tree data files.

Supports all commercial grade C compilers. Requires 128K memory. Version 4.1F.

Lattice

dBC III Plus
List $750 Reg $594 **Sale $499**

With Library Source
List $1500 Reg $1184 **Sale $998**

Use the Lattice dBC III Plus library of functions to write fast C language programs to create, access and update files that are compatible with Ashton-Tate's dBASE III PLUS database management system. dBC III Plus is network ready with functions that solve complicated network database problems. These functions let you lock files or records automatically or manually, prevent you from accidentally locking files or records that are already locked, and allow you to test whether files or records are locked or free. You can share your ISAM files with as many stations as are possible on your network.

Specify compiler (current version): Borland Turbo C, Lattice C, or Microsoft C. Requires 128K memory. Version 1.0.

SoftCraft

Btrieve
List $245 Reg $184 **Sale $169**

Xtrieve
List $245 Reg $184 **Sale $169**

Report Option
List $145 Reg $99 **Sale $89**

Btrieve/N
List $595 Reg $454 **Sale $429**

Xtrieve/N
List $595 Reg $454 **Sale $429**

Report Option/N
List $345 Reg $269 **Sale $249**

Btrieve is a keyed, indexed file management system for use with most programming languages. Btrieve allows a file structure with: record length up to 4K bytes (64K in some environments); up to 24 different keys per file; maximum key size of 255 bytes; a maximum file size of over 4 billion bytes; and file size limited only by physical storage capacity and operating system limitations. Duplicate, modifiable, null and segmented keys are allowed and there is no limit to the number of files open at one time. Written in 8088 assembly language for maximum efficiency, Btrieve uses extensive cache buffering to optimize performance and pre-imaging to automatically recover damaged files. Transaction bracketing and automatic record locking allow you to guarantee the integrity of your data files despite the concurrency problems that arise in a network. The optional Xtrieve is a menu-driven query system that enables Btrieve users to access Btrieve files without writing a program. The Report Option for Xtrieve allows you to easily generate reports.

Specify single-user or multi-user/network version. For multi-user/network version, specify environment: 3Com3plus; IBM TopView; Microsoft Windows; Multilink Advanced; MPOS; Novell Advanced NetWare; XENIX System V/AT; or satellite or server-based IBM PC Network. Btrieve supports most language compilers and interpreters. Requires hard disk and 128K memory (Btrieve routines use 32K). Version 4.10.

CALL TOLL FREE

USA: **800-336-1166**
Canada: **800-225-1166**
Ohio & Alaska (Collect): .. **216-494-3781**

International: **216-494-3781**

Telex: **9102406879**
Easylink: **62806530**

Programmer's Connection
7249 Whipple Avenue NW
North Canton, OH 44720

Please turn the page for our latest price list and ordering information. →

PROGRAMMER'S CONNECTION — QUALITY INTEGRITY Service SUPPORT PRICE

Free Programmer's Connection Buyer's Guide

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

ai - expert systems

	List	Ours
1st-CLASS *by Programs in Motion*	495	399
EXSYS *Development Software by EXSYS*	395	289
EXSYS Runtime System	600	469
LEVEL5 *by Information Builders*	685	569
Logic-Line Series *All varieties by Thunderstone*	CALL	CALL
VP-EXPERT *by Paperback Software* ... *New*	125	79

ai - lisp language

muLISP-87 Interpreter *by Soft Warehouse* .... *New*	250	CALL
muLISP-87 Interpreter & Compiler ..... *New*	350	CALL
Q'Nial *Various by NIAL Systems*	CALL	CALL
Star Sapphire LISP *Compiler by Sapiens*	495	429
TransLISP PLUS *from Solution Systems*	195	125

ai - prolog language

Arity Combination Package	1095	979
Expert System Development Pkg	295	229
File Interchange Toolkit	50	44
PROLOG Compiler & Interpreter	650	569
Screen Design Toolkit	50	44
SQL Development Package	295	229
Arity PROLOG Interpreter	295	229
Arity Standard Prolog	95	77
LPA microPROLOG *All Varieties*	CALL	CALL
MPROLOG Language Primer *LOGICWARE*	50	45
MPROLOG P500 *by LOGICWARE*	495	395
MPROLOG P550 *w/Primer by LOGICWARE*	220	175
Turbo PROLOG *by Borland Intl*	100	64
Turbo PROLOG Toolbox *by Borland Intl*	100	64

ai - smalltalk language

Smalltalk/V	100	84
EGA/VGA Color Option	50	45
Goodies Diskette #1	50	45
Smalltalk/Comm	50	45

ai - texas instruments

Arborist *Decision Tree Software*	595	519
PC Scheme *Lisp*	95	77
Personal Consultant Easy	495	435
Personal Consultant Image	495	435
Personal Consultant Online	995	869
Personal Consultant Plus	2950	2589
Personal Consultant Runtime	95	84

ada language

AdaVantage *GSA-validated by Meridian Software* ...	795	735
AdaVantage Utility Packages	50	47
DOS Environment Package	50	47
Ada *GSA-validated w/maintenance by alsys* ... *New*	3355	3119
Janus/ADA C Pak *by R&R Software*	95	84

apl language

APL*PLUS PC *by STSC*	695	495
APL*PLUS PC Spreadsheet Mgr *by STSC*	195	139
APL*PLUS Tools *by STSC*	295	199
APL*PLUS PS/2 *by STSC*	695	495
ATLAS*GRAPHICS *by STSC*	450	329
Financial/Statistical Library *by STSC*	275	189
Pocket APL *by STSC*	95	69
STATGRAPHICS *by STSC*	895	649

assembly language

386 ASM/LINK *Cross Asm by Phar Lap*	495	389
8088 Assembler *w/Z-80 Translator by 2500 AD*	100	89
ASMLIB *Function Library by BCSoft*	149	125
asmTREE *B-Tree Dev System by BCSoft*	395	329
Cross Assemblers *Various by 2500 AD*	CALL	CALL
EZASM *by C Source*	70	59
Microsoft Macro Assembler	150	CALL
Resident-ASM *by American Software Intl* ..... *New*	159	139
Turbo Debugger *by Speedware*	89	79
Turbo Editasm *by Speedware*	99	84
Visible Computer: 8088 *by Software Masters*	80	64

basic language

db/Lib *for QuickBASIC by AJS Publishing*	139	119
Finally *by Komputerwerk*	99	85
MACH 2 *by Micro Help*	69	55
Microsoft QuickBASIC ..... *New Version*	99	CALL
QBase *Relational Database by Crescent*	89	79
Quick-Tools *by BCSoft*	130	109
QuickPak *by Crescent Software*	69	59
Scientific Subroutine Library *by Wiley*	125	99
Screen Sculptor *by Software Bottling*	125	91
Stay-Res *by MicroHelp*	69	55
True BASIC	100	79
True BASIC *w/Run-time* ..... *Special Price*	150	99
True BASIC 3D Graphics	50	41
True BASIC Developer's Toolkit	50	41
Turbo BASIC Compiler *by Borland Intl*	100	64

blaise products

ASYNCH MANAGER *Specify C or Pascal*	175	135
C TOOLS PLUS/5.0	129	99
KeyPlayer *Super Batch Program*	50	45
LIGHT TOOLS *for Datalight C*	100	65
PASCAL TOOLS & TOOLS 2	175	135
RUNOFF *Text Formatter*	50	45
TURBO ASYNCH PLUS	129	99
TURBO C TOOLS	129	99
TURBO POWER TOOLS PLUS	129	99
VIEW MANAGER *Specify C or Pascal*	275	199

borland products

EUREKA *Equation Solver*	167	105
Reflex: The Analyst	150	99
Sidekick	85	57
Superkey	100	64
Turbo Basic Compiler	100	64
Turbo Basic Database Toolbox	100	64
Turbo Basic Editor Toolbox	100	64
Turbo Basic Telecom Toolbox	100	64
Turbo C Compiler *(Call for support products)*	100	64
Turbo Lightning and Word Wizard	150	94
Turbo Lightning	100	64
Turbo Lightning Word Wizard	70	47
Turbo Pascal	100	64
Turbo Pascal Database Toolbox	100	64
Turbo Pascal Developer's Toolkit	395	289
Turbo Pascal Editor Toolbox	100	64
Turbo Pascal Gameworks Toolbox	100	64
Turbo Pascal Graphix Toolbox	100	64
Turbo Pascal Numerical Methods Toolbox	100	64
Turbo Pascal Tutor	70	41
Turbo Prolog Compiler	100	64
Turbo Prolog Toolbox	100	64

c language

C-terp *by Gimpel Specify compiler*	298	219
C Trainer *with Book by Catalytix*	122	87
DeSmet C *w/Debugger & Large case*	209	184
DeSmet C *w/Debugger only*	159	138
Eco-C88 Modeling Compiler *by Ecosoft* ..... *New*	100	79
Instant C *by Rational Systems*	495	369
Lattice C Compiler *vers. 3.2 from Lattice*	500	265
Mark Williams Let's C *with FREE csd*	75	54
Microsoft C Compiler 5.0 w/CodeView	450	CALL
Microsoft QuickC Compiler	99	CALL
Optimum-C *by Datalight*	139	95
Turbo C Compiler *by Borland*	100	64
Uniware 68000/10/20 Cross Compiler *by SDS*	995	829

c utilities

Blackstar C Library *by Sterling Castle*	125	98
C++ *by Guidelines w/version 1.1 kernel*	195	172
c-tree & r-tree Combo *by FairCom*	650	499
c-tree *ISAM File Manager*	395	289
r-tree *Report Generator*	295	239
Csharp Realtime Toolkit *by Systems Guild*	600	489
Curses Window Dev Pkg *by Aspen Scientific*	119	105
with Source Code	289	249
dBx *dBASE to C Translator by Desktop AI*	350	299
with Source Code	550	419
Entelekon Combo Package ..... *Special Price*	200	125
Flash-up Windows *by Software Bottling*	90	78
GraphiC *Color version by Sci Endeavors*	350	274
HALO *Graphics by Media Cybernetics*	300	205
HALO Development Pkg *for Microsoft*	595	389
The HAMMER *by OES Systems*	195	129
HOOPS *3-D Graphics by Ithaca Software* ..... *New*	575	459
LINK&LOCATE *by Systems & Software* ..... *New*	350	309
PANEL/TC *for Turbo C by Roundhill*	129	95
PANEL Plus *by Roundhill*	495	395
PC Lint *by Gimpel Software*	139	99
Resident-C *by American Software Intl* ..... *New*	150	139
RTC PLUS *Fortran to C by Cobalt Blue*	450	369
Sapiens V8 *Virtual Memory Manager*	300	265
Scientific Subroutine Library *by Wiley*	175	135
TE Developer's Kit *by Sub Systems*	95	85
Vitamin C *by Creative Programming*	225	149
VC Screen *Forms Designer*	100	79
WKS LIBRARY *by Tenon Software* ..... *New*	89	79
Zview *by Data Mgmt Consultants*	245	119

cobol language

COBOL spll *by Flexus*	395	329
FPLIB *for Realia COBOL by BCSoft*	149	129
Micro Focus COBOL *See Micro Focus Section*		
Microsoft COBOL *See Microsoft Section*		
Realia COBOL with RealMENU	1145	899
Realia COBOL	995	783
RealCICS	995	783
RM/COBOL *by Ryan-McFarland*	950	CALL
RM/COBOL 85 *by Ryan-McFarland*	1250	CALL
RM/NET+5 *by Ryan-McFarland*	300	CALL
RM/Screens	395	CALL
SCREENIO *by Norcom*	400	379
screenplay *for COBOL by Flexus*	175	115

database management

Advanced DBMaster *by Macon Software*	*New*	510	419
Clipper *by Nantucket*	*New*	695	CALL
dBASE III Plus *by Ashton-Tate*	*New*	695	389
dbIII Compiler *by WordTech Systems*	*New*	CALL	CALL
dbSQL *by WordTech Systems*	*New*	CALL	CALL
dbxl *by WordTech Systems*	*New*	169	109
dFLOW *by Wallsoft*	*New*	149	119
The Documenter *by Wallsoft*	*New*	295	239
Fox Base Plus	*New*	395	249
Genifer *by Bytel*	*New*	395	275
MAGIC PC *by AKER* ..... *Special Price*	*New*	199	179
Paradox 1.1 *by Ansa/Borland*	*New*	495	359
Paradox 2.0 *by Ansa/Borland*	*New*	725	525
Paradox Network Pack *by Ansa/Borland*	*New*	995	725
Q&A *by Symantec*	*New*	349	219
Quickcode *by Fox & Geller*	*New*	295	179
Quickindex *by Fox & Geller*	*New*	149	95
Quickreport *by Fox & Geller*	*New*	295	179
QuickSilver *by WordTech Systems*	*New*	599	349
R:Base 5000 *by Microrim*	*New*	495	CALL
R:Base System V *by Microrim*	*New*	700	CALL
\rdb *by Robinson-Shafer-Wright*	*New*	139	119
Advanced Revelation *by COSMOS*	*New*	950	CALL
SQL Base *by Gupta Technologies*	*New*	995	CALL
Multi-User Version	*New*	1995	CALL
Tom Rettig's Library *by Tom Rettig & Assoc* ...	*New*	100	79
UI Programmer *by Wallsoft*	*New*	295	239
VP-INFO *by Paperback Software*	*New*	125	79
VP-PLANNER *by Paperback Software*	*New*	100	63
VP-PLANNER PLUS *by Paperback Software* ...	*New*	125	79
XDB II *by Software Systems Technology*	*New*	395	CALL
C Programming Interface	*New*	295	CALL

debuggers & profilers

386 DEBUG *Cross Debugger by Phar Lap*	195	129
Advanced Trace-86 *by Morgan Computing*	175	115
Codesmith-86 *by Visual Age*	145	98
DSD87 *by Soft Advances*	125	79
MiniProbe *by Atron*	395	369
Periscope I *with Board by Periscope*	345	275
Periscope II *with NMI Breakout Switch*	175	139
Periscope II-X *Software only*	145	105
Periscope III *8 MHz version*	995	795
Periscope III *10 MHz version*	1095	875
The PROFILER *with Source Code by DWB*	125	89
TURBOsmith *Source debugger for Turbo Pascal*	99	69
The WATCHER *Profiler by Stony Brook*	60	51

disk utilities

Back-It *by Gazelle Systems*	130	115
Disk Optimizer *by Softlogic Systems*	60	55
Disk Technician *by Prime Solutions*	100	89
FASTBACK *by 5th Generation Systems*	159	115
FASTBACK PLUS *by 5th Generation Systems* .. *New*	189	139
United Software Security *Take Two Manager* .. *New*	139	119
Vcache *by Golden Bow Systems*	50	47
Vopt *by Golden Bow Systems*	50	47
Vfeature *by Golden Bow Systems*	80	74
Vfeature Deluxe *by Golden Bow Systems*	120	111
XenoCopy-PC *by XenoSoft*	80	69

dos utilities

Advanced Norton Utilities	150	89
Command Plus *by ESP Software*	80	69
Desqview *from Quarterdeck*	130	115
FANSI-CONSOLE *by Hersey Micro*	75	62
Mace Utilities *Paul Mace Software*	99	89
MicroHelp Utility *by MicroHelp*	59	49
Norton Commander *by Peter Norton*	75	55
Norton Utilities *by Peter Norton*	100	59
OPAL *Shell Language by Software Factory*	99	89
Q-DOS II *by Gazelle Systems*	70	59
Taskview *by Sunny Hill Software*	80	55

essential products

Breakout Debugger *by Essential Software*	125	89
C Utility Library	185	119
Essential Communications	185	125
Essential Communications *with Break Out*	250	189
Essential Graphics	250	183
/*residentC*/ ..... *New*	99	85
with Source Code ..... *New*	198	169
ScreenStar ..... *New*	99	85
with Library Source Code ..... *New*	198	169

forth language

CFORTH *Native Code Compiler by LMI*	300	229
FORTH/83 Metacompiler *Specify Target*	750	599
MMS Forth *by Miller Microcomputer Svcs* ..... *New*	180	159
PC/FORTH *by Laboratory Microsystems*	150	109
PC/FORTH+ *by Laboratory Microsystems*	250	199
Programmer's Package #1 *by LMI*	250	199
Programmer's Package #2 *by LMI*	350	279
Programmer's Package #3 *by LMI*	500	399
UR/FORTH *Also Available for OS/2 by LMI*	350	279
UR/FORTH Libraries	500	395

fortran language

50 MORE: FORTRAN *by Peerless Scientific*	125	95
ACS Time Series *by Alpha Computer Service*	495	389
AUTOMATED PROGRAMMER *by KGK Automated* ..	995	949
Essential Graphics *by Essential Software*	250	183
Forlib-Plus *by Alpha Computer Service*	70	44
FORTRAN Addenda *by Impulse Engr*	165	138
HALO *Graphics by Media Cybernetics*	300	205
I/O PRO w/No Limit Library *by MEF*	250	219
Microcompatibles Combo Package	240	215
Grafmatic	135	117
Plotmatic	135	117
Microsoft FORTRAN *w/CodeView*	450	CALL
No Limit Library *by MEF Environmental*	129	109
Numerical Analyst *by MAGUS*	295	199
PANEL *by Roundhill Computer Systems*	295	199
RM/FORTRAN *by Ryan-McFarland*	599	399
Scientific Subroutine Library *by Wiley*	175	135
Statistician *by Alpha Computer Service*	295	235
STATLIB.GL: *by Wiley*	295	239
STATLIB.TSF: *by Wiley*	295	239
Strings & Things *by Alpha Computer Service*	70	45

greenleaf products

Greenleaf C Sampler *Specify QuickC or Turbo C*	95	69
Greenleaf Comm Library	185	125
Greenleaf Data Windows Library	225	155
with Source Code	395	249
Greenleaf Functions	185	125

help utilities

HELP/Control *by MDS*	125	99
On-line Help *from Opt-Tech*	149	99
SoftScreen/HELP *by Dialectic Systems*	195	149

lattice products

Lattice C Compiler *ver 3.2 from Lattice*	500	265
with Library Source Code	900	495
C Cross Reference Generator	50	37
with Source Code	200	139
C-Food Smorgasbord *Function Library*	150	95
with Source Code	300	179
C-Sprite *Source Level Debugger*	175	CALL
Curses *Screen Manager*	125	85
with Source Code	250	169
dBC III	250	169
with Source Code	500	356
dBC III Plus	750	499
with Source Code	1500	998
LMK *Make Facility*	195	138
RPG II Combo *All three items below*	1100	875
RPG II Compiler *No Royalties*	750	625
SEU *Source Entry Utility*	250	199
Sort/Merge	250	199
Screen Design Aid Utility *for RPG II*	350	309
SecretDisk II *Encryption Utility*	79	59
SideTalk *Resident Communications*	120	88
SSP/PC *Scientific Subroutine Library*	350	269
Text Management Utilities	120	88

metagraphics products

FontWINDOW	95	79
LightWINDOW/C *for Datalight C*	95	79
MetaWINDOW *No Royalties*	195	159
MetaWINDOW/PLUS	275	229
TurboWINDOW/C *for Turbo C*	95	79
TurboWINDOW/Pascal *for Turbo Pascal*	95	79

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

micro focus products

Micro Focus COBOL/2	900	729
Micro Focus COBOL/10 *Ad hoc Report Writer*	495	395
Micro Focus COBOL/10 *for DOS 3.X Networks*	995	795
Micro Focus FORMS-2	295	235
Micro Focus Level II COBOL *w/Animator*	495	395
Level II COBOL	349	279
Level II Animator	195	155
Micro Focus PC-CICS	1495	1189
with **Micro/SPF**	1595	1269
Micro Focus Personal COBOL	149	119
Micro Focus Professional COBOL	CALL	CALL
Micro Focus SOURCEWRITER	995	795
Micro Focus VS COBOL/XENIX	1495	1195

microport products

386 Unlimited License Kit	249	209
AT Unlimited License Kit	249	209
DOSMerge286 *Specify 2-Users or Unlimited*	149	129
DOSMerge386 *2-Users*	395	345
DOSMerge386 *Unlimited Users*	495	429
System V/386 Combination	799	669
386 Runtime System	199	169
386 Software Development System	499	429
Text Preparation System	199	169
System V/AT Combination	549	465
AT Runtime System	199	169
AT Software Development System	249	209
Text Preparation System	199	169
	395	295

microsoft products

Microsoft BASIC Compiler *for XENIX*	695	CALL
Microsoft BASIC Interpreter *for XENIX*	350	CALL
Microsoft C Compiler 5.0 *w/CodeView*	450	CALL
Microsoft COBOL Compiler *with COBOL Tools*	700	CALL
for XENIX	995	CALL
Microsoft Excel *New*	495	CALL
Microsoft FORTRAN *Optimizing Compiler*	450	CALL
Microsoft FORTRAN *for XENIX*	695	CALL
Microsoft Learning DOS	50	CALL
Microsoft Macro Assembler	150	CALL
Microsoft Mouse *Specify Serial or Bus*	150	CALL
with Microsoft Windows	200	CALL
with EasyCAD	175	CALL
Microsoft Pascal Compiler	300	CALL
for XENIX	695	CALL
Microsoft QuickBASIC	99	CALL
Microsoft QuickC	99	CALL
Microsoft Windows	99	CALL
Microsoft Windows 386 *New*	195	CALL
Microsoft Windows Development Kit	500	CALL
Microsoft Word	450	CALL
Microsoft Works *New*	195	CALL

mks products

MKS AWK	75	65
MKS RCS *Revision Control System*	189	155
MKS Toolkit *with MKS VI Editor*	139	109
MKS Trilogy *with AWK, CRYPT & Korn Shell*	119	99
MKS VI *Editor by MKS*	75	65

modula-2 language

LOGITECH Modula-2 Development System	249	199
Modula-2 Compiler Pack	99	79
Modula-2 Toolkit	169	139
LOGITECH Modula-2 Window Pkg	49	39
Macro2 *Macro preprocessor by PMI*	89	79
ModBase *by PMI*	89	79
ModGraph *by TEGNA*	50	45
MODULA-2 *by Stony Brook* *New*	195	169
MODULA-2 with Utilities *by Stony Brook* *New*	345	299
Repertoire *by PMI*	89	75
Science & Engrg Tools *by Quinn-Curtis*	75	67
Universal Graphics Library *by Quinn-Curtis*	130	119

mouse products

LOGIMOUSE BUS *with PLUS Pkg by LOGITECH*	119	98
with PLUS & PC Paintbrush	149	119
with PLUS & CAD Software	189	153
with PLUS & CAD & Paint	219	179
with PLUS & First Publisher	CALL	CALL
LOGIMOUSE C7 *with PLUS Pkg, Specify Connector*	119	98
with PLUS & PC Paintbrush	149	119
with PLUS & CAD Software	189	153
with PLUS & CAD & Paint	219	179
with PLUS & First Publisher	CALL	CALL
Microsoft Mouse *See Microsoft Section*		

other languages

ACTOR *by Whitewater Group*	495	419
CCS MUMPS *All varieties by MGlobal*	CALL	CALL
Marshal Pascal *by Marshal Language Systems*	189	155
Pascal-2 *by Oregon Software*	395	269
Personal REXX *by Mansfield Software*	125	99
SNOBOL4+ *by Catspaw*	95	80

other products

Carbon Copy Plus *by Meridian Technology*	195	159
Dan Bricklin's Demo Pgm *by Software Garden*	75	57
Dan Bricklin's Demo Tutorial	50	45
Fast Forward *by Mark Williams*	70	59
Instant Replay *by Nostradamus*	150	CALL
MicroTEX *Typesetting from Addison-Wesley*	295	CALL
Printer Drivers	CALL	CALL
muMATH *by Soft Warehouse* *New*	300	199
Net-Tools *by BCSoft*	149	129
Norton Guides *Specify Language*	100	65
OPT-Tech Sort *by Opt-Tech Data Proc*	149	99
PC-MOS/386 *by The Software Link* *New*	195	179
PC/TOOLS *by Custom Software Systems*	49	45
Resident Expert *Specify lang by Santa Rita* *New*	59	55
Screen Machine *by MicroHelp*	79	59
SuperSort *by LifeStyle*	139	119

phoenix products

Pasm86 *Macro Assembler version 2.0*	195	188
Pdisk *Hard Disk & Backup Utility*	145	99
Pfantasy Pac *Phoenix Combo*	995	595
Pfinish *Execution Profiler*	395	209
Pfix86plus *Symbolic Debugger*	395	209
PforCe *Specify C Compiler*	395	209
PforCe++ *Specify C Compiler and C++*	395	209
Plink86plus *Overlay Linker*	495	275
Pmaker *Make Utility*	125	78
Pmate *Macro Text Editor*	195	108
Pre-C *Lint Utility*	295	154
Ptel *Binary File Transfer Program*	195	108

polytron products

PolyBoost *Software Accelerator* *Special Price*	80	54
PolyDesk III	99	72
PolyDesk III Archivist	50	42
PolyDesk III Cryptographer	50	42
PolyDesk III Talk	70	52
PolyLibrarian *Library Manager*	99	89
PolyLibrarian II *Library Manager*	149	129
PolyMake *UNIX-like Make Facility*	149	129
PolyShell *Special Price*	149	105
Polytron C Beautifier	50	45
PolyXREF *Complete Cross Ref Utility*	219	185
PolyXREF *One language only*	129	109
PVCS Corporate *Version Control System*	395	329
PVCS Personal	149	129

program mgmt utilities

Interactive EASYFLOW *by Haveatree*	150	125
PrintQ *by Software Directions*	89	84
Quilt Computing Combo *QMake & SRMS*	250	199
Sapiens MAKE	179	155
Sapiens MAKE & V8	439	379
Source Print *by Aldebaran Labs*	97	75
TLIB *Version Control System by Burton*	100	89
Tree Diagrammer *by Aldebaran Labs*	77	67

sco products

Complete XENIX System V *by SCO*	1295	994
Development System	595	499
Operating System *Specify XT or AT*	595	499
Text Processing Package	195	144
Lyrix *by SCO*	595	449
SCO Professional *1-2-3 Workalike for XENIX*	795	595
SCO XENIX-NET	595	495
XENIX System V 386 *by SCO*	CALL	CALL

softcraft products

Btrieve *ISAM Mgr with No Royalties*	245	169
Xtrieve *Query Utility*	245	169
Report Option *for Xtrieve*	145	89
Btrieve/N *for Networks*	595	429
Xtrieve/N	595	429
Report Option/N *for Xtrieve/N*	345	249
XQL *New*	795	CALL

text editors

Brief & dBrief Combo *from Solution Systems*	275	CALL
Brief	195	CALL
dBrief *Customizes Brief for dBASE III*	95	CALL
de *by David Livshin*	75	65
Epsilon *Emacs-like editor by Lugaru*	195	147
KEDIT *by Mansfield Software*	125	99
Micro/SPF *by PHASER SYSTEMS*	175	139
Microsoft Word	450	CALL
PC/VI *by Custom Software Systems*	149	99
SPF/PC *by Command Technology Corp*	CALL	CALL
Vedit Plus *by CompuView*	185	128

turbo pascal utilities

ALICE *Interpreter by Software Channels*	95	66
AZATAR DOS Toolkit *by AZATAR* *New*	95	85
DOS/BIOS & Mouse Tools *by Quinn-Curtis*	75	67
Flash-up *by Software Bottling* *New*	89	78
Flash-up Developer's Toolbox *New*	49	45
MACH 2 *for Turbo Pascal by Micro Help*	69	55
MetraByte D/A Tools *by Quinn-Curtis*	100	89
Science & Engrg Tools *by Quinn-Curtis*	75	67
Screen Sculptor *by Software Bottling*	125	91
Speed Screen *by Software Bottling*	35	32
System Builder *by Royal American*	150	129
IMPEX *Query Utility*	100	89
Report Builder	130	115
TDebugPLUS *by TurboPower Software*	60	49
Tmark *by Tangent Designs*	80	69
Turbo Professional *from TurboPower* *New Version*	99	79
TurboHALO *from tMSI*	95	75
TurboPower Utilities *by TurboPower*	95	78
TurboRef *by Gracon Services*	50	35
TURBOsmith *Source Debugger by Visual Age*	99	69
Universal Graphics Library *by Quinn-Curtis*	130	119

wendin products

Operating System Toolbox	99	79
PCNX *Operating system*	99	79
PCVMS *Similar to VAX/VMS*	99	79
Wendin-DOS *Multitasking DOS*	99	85
XTC *Text Editor w/Pascal source*	99	75

xenix/unix products

Btrieve *ISAM File Mgr by SoftCraft*	595	429
C-terp *by Gimpel*	498	379
c-tree *ISAM Mgr by FairCom*	395	289
dBx *with Library Source by Desktop AI*	550	419
DIRECTORY SHELL 286 *by American Mgmt Sys*	349	295
DIRECTORY SHELL 386 *by American Mgmt Sys*	495	415
Epsilon *Text Editor by Lugaru* *New*	195	147
Micro Focus Products *See Micro Focus Section*		
Microport Products *See Microport Section*		
Microsoft Products *See Microsoft Section*		
PANEL Plus *by Roundhill Computer Systems*	795	535
REAL-TOOLS *Binary Version by PCT*	99	89
Complete Source Version	999	729
RM/COBOL *by Ryan-McFarland*	1250	949
RM/FORTRAN *by Ryan-McFarland*	750	549
SCO Products *See SCO Section*		

Terms are subject to change.

©1987 Programmers Connection, Inc.

LOWEST PRICES

Due to printing lead times, some of our current prices may differ from those shown here. Call for latest pricing.

FREE SHIPPING

Orders within the USA (including Alaska & Hawaii) are shipped FREE via UPS. Express shipping is available at the shipping carrier's standard rate with no rush fees or handling charges. To avoid delays when ordering by mail, please call first to determine the exact cost of express shipping.

CREDIT CARDS

VISA and MasterCard are accepted at no extra cost. Your card is charged when your order is shipped. Mail orders please include credit card expiration date and authorized signature.

CODs AND POs

CODs and Purchase Orders are accepted at no extra cost. No personal checks are accepted on COD orders. POs with net 30-day terms (with initial minimum order of $100) are available to qualified US accounts only.

SALES TAX

Orders outside of Ohio are not charged state sales tax. Ohio customers please add 6% Ohio tax or provide proof of tax-exemption.

INTERNATIONAL ORDERS

Shipping charges for International and Canadian orders are based on the shipping carrier's standard rate. Since rates vary between carriers, please call or write for the exact cost. International orders (except Canada), please include an additional $10 for export preparation. All payments must be made with US funds drawn on a US bank. Please include your telephone number when ordering by mail. Due to government regulations, we cannot ship to all countries.

VOLUME ORDERS

Volume orders may qualify for additional discounts. Call us for special pricing.

SOUND ADVICE

Our knowledgeable technical staff can answer technical questions, assist in comparing products and send you detailed product information tailored to your needs.

30-DAY GUARANTEE

Most of our products (excluding books) come with a 30-day documentation evaluation period or a 30-day return guarantee. Please note that some manufacturers restrict us from offering guarantees on their products. Call for more information.

MAIL ORDERS

Please include your telephone number on all mail orders. Be sure to specify computer, operating system and any applicable compiler or hardware interface(s). Send mail orders to:

Programmer's Connection
7249 Whipple Ave. NW
North Canton, OH 44720

USA	**800-336-1166**
CANADA	**800-225-1166**
OHIO & ALASKA (Collect)	**216-494-3781**
TELEX	**910240879**
EASYLINK	**62806530**
INTERNATIONAL	**216-494-3781**
CUSTOMER SERVICE	**216-494-8899**

Hours: Weekdays 8:30 AM to 8:00 PM EST.

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

Code from "Discovering..." by Tom Swan

1

Listing 1: ENV.PAS

```
PROGRAM Env;

{ Demonstrate how to read environment strings in Turbo Pascal 4.0.
  (C) 1987 by Swan Software, P.O. Box 206, Lititz PA 17543 }

TYPE

   CharPtr = ^Char;   { Pointer to characters }

VAR
   s : String;   { Holds environment string settings }


PROCEDURE AdvanceCP( VAR p : CharPtr );

{ Add one to p's offset address.  This procedure cannot advance
  p beyond a segment boundary.  Later Turbo Pascal versions might
  let you replace calls to AdvanceCP(p) with Inc(p). }

BEGIN
   p := Ptr( Seg(p^), Ofs(p^) + 1 )
END; { AdvanceCP }


FUNCTION GetEnv( s : String ) : CharPtr;

{ Search environment for string s and return a pointer to the
  setting for this environment variable.  If the variable is not
  found, return nil.  For example, if s equals 'COMSPEC', then
  GetEnv returns a pointer to the current COMSPEC setting,
  probably C:\COMMAND.COM.  String s is case sensitive.  In most
  cases, it should be in all caps. }

VAR

   p : CharPtr;   { Pointer to environment characters }


   FUNCTION Match( VAR s : String; VAR p : CharPtr ) : Boolean;

   { True if characters at p^ match string s exactly.  If they do
     match, then p^ addresses the character after the end of the
     matching string.  If they do not match, then p addresses the
     character that does not match. }

   VAR

      i : Integer;   { FOR-loop control variable }

   BEGIN
      FOR i := 1 TO Length(s) DO   { Examine all chars in s }
      BEGIN
         IF p^ <> s[i] THEN        { If mismatch found... }
         BEGIN
            Match := FALSE;        { ... return false }
            exit                   {     and exit Match }
         END; { if }
         AdvanceCP( p )            { Continue looking }
      END; { for }

    { All characters match.  Return true if character
      at p is an equal sign. }

      Match := ( p^ = '=' )

   END; { Match }


BEGIN { GetEnv }

{ Pick up environment pointer at PSP:002C, and assign to p making
  it point to the first character of the environment block. }
```

2

```
   p := Ptr( MemW[ PrefixSeg:$002C ], 0 );

   WHILE p^ <> Chr(0) DO    { While not at end of environment block }
   BEGIN
      IF Match( s, p )      { If the string s matches chars at p^ }
       THEN
         BEGIN
            AdvanceCP(p);   { Advance p past = sign }
            GetEnv := p;    { Return pointer to ASCIIZ string }
            exit            { Exit procedure early }
         END; { if }
      WHILE p^ <> Chr(0) DO   { Find end of ASCIIZ string }
         AdvanceCP( p );
      AdvanceCP( p )        { Advance p to next string (if any) }
   END; { while }

   GetEnv := NIL  { Environment variable not found }

END; { GetEnv }


PROCEDURE ASCIIZtoStr( p : CharPtr; VAR s : String );

{ Convert ASCIIZ character string addressed by p into a length-byte
  string variable, s.  The character array must end in a null zero
  byte.  The string is limited to 255 characters.  If p is NIL, then
  ASCIIZtoStr returns zero for the string length. }

VAR

   count : integer;   { String length counter }

BEGIN

   count := 0;  { Initialize string length to zero }

   IF p <> NIL THEN  { Return zero length string for nil pointer }

      WHILE ( p^ <> Chr(0) ) AND ( count < 255 ) DO
      BEGIN
         count := count + 1;   { Count characters transferred }
         s[ count ] := p^;     { Assign char at p to string s }
         AdvanceCP( p )        { Advance pointer to next char }
      END; { while }

   s[0] := Chr( count )       { Assign string length byte }

END; { ASCIIZtoStr }


BEGIN
   Writeln;
   Writeln( 'Test reading environment strings' );
   Writeln;
   ASCIIZtoStr( GetEnv( 'COMSPEC' ), s );
   writeln( 'COMSPEC ..... ', s );
   ASCIIZtoStr( GetEnv( 'PATH' ), s );
   writeln( 'PATH ........ ', s );
   ASCIIZtoStr( GetEnv( 'PROMPT' ), s );
   writeln( 'PROMPT ...... ', s );
   ASCIIZtoStr( GetEnv( 'TEMP' ), s );
   writeln( 'TEMP ........ ', s )
END.
```


PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

vLIB

the user interface library for C

A comprehensive, easy to use library of custom C functions for building sophisticated PC applications.

Windows • *overlapping, tiled, built-in window manager*

Menus • *vertical, horizontal ("Lotus" style), pulldown ("Mac" style)*

Forms • *single call input and display, multiple field types*

Pop Ups • *messages, prompts, selection lists*

Full Editing Input

High Speed Display

... and much, much more!

Give your programs a professional look! Develop applications faster!

Compilers supported: Microsoft C and Quick C, Borland Turbo C, Lattice C, and Datalight C.

All memory models supported.

Demo disk with usable sample programs, source, and reference card . . . **$10**

Library with reference card and detailed programmer's manual **$99**

Library (as above) with full source code **$149**

Prices include shipping, CA residents add 7% sales tax.

No royalties. Site license available.

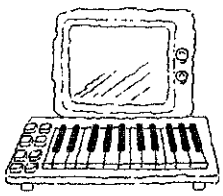
Call or write today:

Pathfinder Associates
291 Madrone Avenue
Santa Clara, CA 95051
(408) 984-2256

Visa and MasterCard accepted.

Number 145 on the Reader Service Card

Now you can play Assembler in the key of C. risC™ from IMSI.™



Sometimes you've just *got* to write in Assembler. It's tedious. But it makes the machine perform.

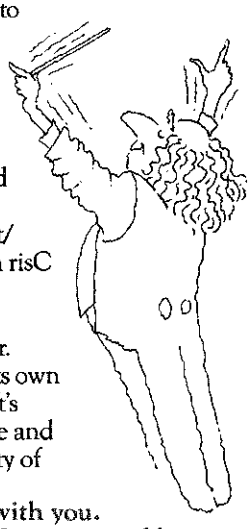
When you're used to writing in a friendlier and easier language like C, working in Assembler is a little like working in Greek. But now, there is a better way to write Assembler code. Add unbelievable speed to program development. And make programs easier to maintain. It's called risC.

risC transposes for you. Fast. risC is the first portable, C-like, object-oriented, High-level Assembly Language (HAL). It includes features of object-oriented high-level languages like Smalltalk, Objective C, LISP and PROLOG.

You write in a C-like syntax, and risC transposes to Assembler. At Assembler speeds. With tight Assembler precision.

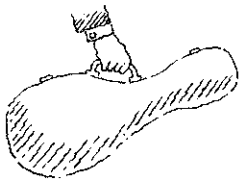
Your objects will work in concert. With risC, your program costs will go down, because you can create objects and operators to go with them. And risC contains a complete object-oriented messaging kernel (source code included) which allows risC objects (.EXE files) to pass messages back and forth.

Add your own personal touch. You can tailor the language to your own personal style. risC's flexible syntax allows you to create customized compilers thanks to its language extension capabilities ("packages"). risC keeps "packages" in compiled form for speedy compilation times. Your development process will go faster than ever before.



Your program development costs will be reduced even more because risC allows you to specify the exact Assembler code generated by each object/operator combination. Producing .ASM files with risC variable names and comments intact.

risC helps you identify when you're off key. You'll have better applications. Faster and cheaper. Because risC allows debugging two ways: under its own source code debugger, DBG, and under Microsoft's CodeView. To bring down your development time and costs even more, risC interfaces with a large variety of existing .OBJ library routines.



It's easy to take with you. Unlike Assembler, risC creates portable programs. So you can easily port your applications to other current and future machine architectures.

risC is a sophisticated programming tool. risC has compiler options allowing you to interface with many different C and Pascal compilers and with different 8086 models—'NEAR,' 'FAR,' and 'INTERRUPT.' So if you're a serious programmer, you'll find risC is a serious programming tool. It's just easier.

Microsoft and CodeView are registered trademarks of Microsoft Corporation.

Order today. And play Assembler in the key of risC.

Only $79.95, with a 30-day, money-back guarantee. In CA add 6% sales tax.

To order risC, just call IMSI at **(415) 454-7101,** or call toll-free, **1-800-222-4723.** (In CA call **1-800-562-4723).** If you prefer, return this coupon with your credit card #, or a check for $79.95 to IMSI, 1299 4th Street, San Rafael, CA 94901. Please add $3.00 shipping and handling.

Name_____ Title_____

Firm_____

Address_____

City_____ State_____ Zip_____

☐ Visa® ☐ MasterCard® _____ exp. date_____

Signature_____

PJ188

risC by IMSI™

Number 159 on the Reader Service Card

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

Time Is on My Side . . . Yes It Is

by
Frank D. Greco

The Time Is Now

In the course of fine-tuning, it's occasionally necessary to get a performance profile of your code. This could be done in several ways. One way is to statistically sample the instruction pointer and map the instruction addresses to function calls through your link map. Another is to embed counters after each statement in your source code (obviously, through some mechanized technique). Most of the time, though, you just need a rough "feel" for the program's performance at certain critical sections. Wouldn't it be nice if you could insert a little software probe into your source code during your performance tuning that would make timing statistics available? And, to avoid outputting statistics on your display (that is, standard out—stdout), how about sending these statistics through the standard error channel (stderr, that is), which is redirectable ("Redirection, or They Went That-a-way," *PJ*, Vol. 5.1)? Think I'm leading up to something? But of course.

Let's look at how the PC keeps time, how we can get the time from the PC, and how to write a program that uses these time values to obtain some timing statistics.

Lunch Time

What the programmer must do to time a function call or an occasional block of C code is fairly obvious. The piece of code to be timed is "sandwiched" by two function calls that obtain the current value of the clock. Subtracting the first clock value from the second will yield the elapsed time. In pseudocode this would be

```
Time1 <-- CurrentClockTick
Execute the Function (or Block of code)
Time2<-- CurrentClockTick

Display (Time2 - Time1)
```

What Time Is It?

So far, so good. First and foremost, we need a method of obtaining a somewhat accurate value of the current time. The PC family keeps a sense of time by use of a timer chip (typically an Intel 8253). This chip counts from 0 up to a specific value (65,536, for example), then emits a pulse to the interrupt controller (Intel 8259A, for example). This process causes an interrupt 8—the clock tick—to occur, which increments a counter (zeroed at midnight) somewhere in the BIOS data area. Now, we can go into the PC's low memory and pull this counter out directly, but that's not very clean. Besides, we should be mentally preparing ourselves, whether we like it or not, for (trumpets please . . .) OS/2, where logical data segments can possibly "float" all around physical memory. Thankfully, there is a BIOS call to help us out.

Listing 1—**gettime.c**—is a small function written in Microsoft C that uses the BIOS to snag the current time in clock ticks. Interrupt 1Ah allows the programmer to set (AH = 1) or get (AH = 0) the current time. When getting the time, this interrupt returns the high order bits of the current time in the CX register. The remainder, that is, the low order bits, are left in DX. In addition, if the AL register is zero, midnight did not occur. If it is non-zero, midnight did indeed pass and it is the responsibility of the programmer to correctly adjust the timer values by 24 hours' worth of clock ticks.

Is That the Right Time?

You can verify that this function does indeed work by writing a small main function that calls **gettime()** and displays the current clock tick (Listing 2—gettix.c). You can verify that this function does indeed work correctly by comparing the timer tick value displayed by your program with the 4 byte counter in the BIOS (pssst . . . it's kept at hex 0040:006C). In our **gettix.c** program, I used a far pointer to access the counter directly.

Notice that because the granularity of our timing is not extremely fine (that is, precise to the nearest 1/18 second), we get the same value. You would expect that the second time we looked at the clock, it would be longer. For our performance tuning purposes (and for most purposes), the timing resolution really does not have to be that fine. Figure 1 shows some sample runs of **gettix.c:**

```
C:\> gettix
BIOS: 944408  Thru far ptr: 944408

C:\> gettix
BIOS: 944496  Thru far ptr: 944496

C:\> gettix
BIOS: 944584  Thru far ptr: 944584
```

Figure 1: Sample runs of gettix.c

Making Time

Once you are convinced that the function **gettime()** works, all you have left to do is to implement the pseudocode that calculates rough timing statistics. To make life easier, it would be nice if all you had to do was say,

```
TIME(thing);
```


PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

This would "sandwich" the calls to **gettime()**, subtract the values obtained, convert the timer ticks to seconds (because, after all, we are human), and display the result through the standard error channel. Also, how about a feature that allows you to selectively turn on timing or turn off timing during performance tests? Too much to ask? Not really.

Holy Macro

The underused C preprocessor comes in very handy for this application. Let's invent some useful macros for our pseudocode. Please note that I have capitalized all my macro names. I usually do this so that I can differentiate macro names from my variable names at a glance. It's not necessary, but it's very helpful during those long, tedious debugging sessions at 3 a.m., when you're running on fumes and your eyes can't tell the difference between **CountClick** and **CountTick**.

```
Time1 = gettime()
```

becomes

```
START()
```

and

```
Time2 = gettime()
```

becomes

```
STOP()
```

If you write programs for a living, you understand what it means to make your system "robust" and "crash proof."

Now we need to come up with a method of converting the difference of these timer ticks into seconds and sending them through the standard error channel (**stderr**):

```
fprintf(stderr, "%lf\n", TIX_DIFF/TIX_PER_SEC)
```

becomes

```
PRINT_TIME()
```

Note that TIX_DIFF represents the difference between Time1 and Time2 and that TIX_PER_SEC is a manifest constant representing 18.20648 clock ticks per second for the PC.

Now our section of code looks like

```
START();
Code to be timed;
STOP();
PRINT_TIME();
```

We can simplify this even further, if we wish, with another macro:

```
#define TIME(code)   START();code;STOP();\
PRINT_TIME();
```

Changing the TIME

One small refinement is left before our masterpiece macro is finished. This macro, as it sits, merely displays a time value. To make the results more understandable, we could annotate each standard error output with an arbitrary string. That makes our final macro,

```
#define TIME(code,string) START();\
code;STOP();PRINT_TIME(string)
```

In order for this macro to work, we must change the macro definition for PRINT_TIME() to

```
#define PRINT_TIME(string) fprintf(stderr,
"%s: %lf\n",string,TIX_DIFF/TIX_PER_SEC)
```

An obvious application of this cute macro is a timing "shell" that determines how many seconds a program took in execution. Listing 3 and Listing 4, **timer.h** and **timer.c**, demonstrate how such a program could be written.

Line 40 is the crux of the entire program and demonstrates our completed TIME() macro in action. Notice that we are timing the expression,

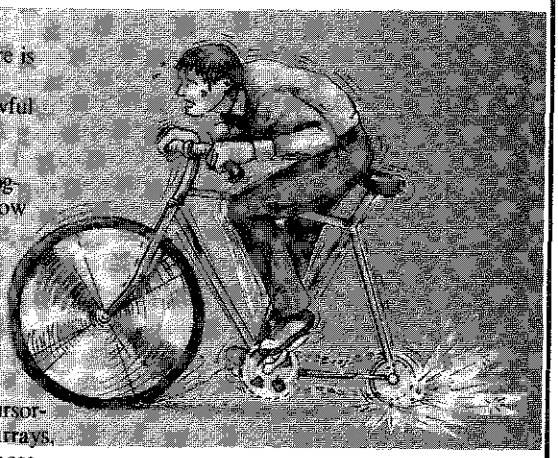
```
ret = spawnvp(P_WAIT, bp, ++argv)
```

which suspends (whattaya expect . . . it's MS-DOS!) the parent process, timer, and spawns a process based on the first argument on the command line with the remainder of the arguments being the args to the spawned process. In addition, the return code for **spawnvp()** is saved in the variable **ret**.

What should be noted is that according to the Microsoft documentation, if **spawnvp()** notices that the program requested to be spawned is not in MS-DOS executable format, it returns the error code represented by ENOEXEC. Unfortunately, this doesn't work too well, or at least not as one would expect [I haven't tried to run an .EXE file that has some garbage in its EXE header—FG]. If you attempt to run an obviously invalid executable (**timer.c**, for example), **spawnvp()** will try to run it and subsequently crash your machine. If you write programs for a living, you understand what is meant by making your system "robust" and "crash-proof." To make this program truly robust, you would have to check the file extension of the program to be spawned and verify that it was an EXE or COM file. This is exactly how **command.com** prevents a user from running a non-executable file. This, however (of course), involves a little bit of work (perhaps grist for a future article) and actually may not always work. If a user renames a C source file, for example, to a binary image file and attempts to run it, the machine will give him

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

If You Have Turbo C You Have Half Your C-Programming Vehicle



Turbo C is a great compiler but there is one vital cog missing—debugging. Without it, you have to spend an awful lot of energy to go a short distance.

Gimpel Software's C-terp, long recognized as the leading C interpreter, now fully supports Turbo C with complete compatibility guaranteed.

Interactive Debugger—Our debugging facilities include split screen (code in upper portion, dialog in lower), breakpoints (sticky, temporary, line/function, cursor-directed), display of structures and arrays, execution of any expression (even those involving macros), function traceback with arguments, watch expressions and watch conditions (watchpoints). Our watch expressions can be structs or arrays. We catch out-of-bounds pointers!

No Toy—Full K&R with ANSI enhancements. Multiple-module with a built-in automatic make. It has virtual memory option (with optional direct use of extended memory) and a shared symbol option for those big programs. It supports graphics, dual displays and the EGA 43-line mode.

Links to external libraries—(both code and data, automatically) which can call back to interpreted functions. Function pointers are compiler compatible.

100% Turbo-C compatible.—Same header (.h) files, data alignment, bit field orderings and preprocessor variables as your compiler. We link in your compiler's library.

Our reconfigurable editor—is multifile and comes with a configuration script to mimic Turbo's editor.

The missing wheel that will turn your half-cycle into a bicycle

C-terp

Order C-terp today!

Call (215) 584-4261

Introductory Price for Turbo C-terp:
$139.00

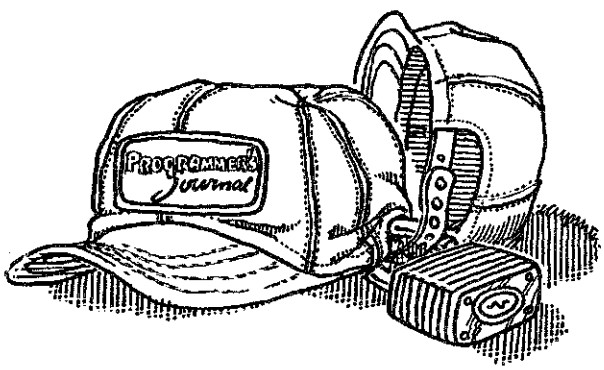
VISA, MC, COD—30 day money back guarantee

C-terp Version 3.0 is also available for the following compilers:
Microsoft, Lattice, Aztec, C86, and Mark Williams ($298) and Xenix ($498).

GIMPEL SOFTWARE
3207 Hogarth Lane
Collegeville, PA 19426

C-terp is a trademark of Gimpel Software, and Turbo C of Borland International.

Contact this Advertiser directly



-CAPSLOCK-

or her the cold shoulder. Good ol' MS-DOS. I can't believe it's 1987 and things like this happen!

Try this if you don't believe me:

```
C:\>  rename timer.c pgm.com
C:\>  pgm
```

It's outer space time. Before you try the above, I strongly suggest you save your RAM-disk files to your winchester.

TIMER Errors

But you say that the standard error output stream is still connected to your console and interferes with the timed-program's output. Well, in Vol. 5.1 of *Programmer's Journal*, I developed a small program (**err.c**) that allows redirection of standard error. If you run the program to be evaluated via my err utility, all the performance timings can be redirected and saved in a file. The following are two examples of how you would use **err**:

```
C:\> err "timer ls -l"  errfile
```

Time a program (ls -l) and save the timing statistics in the file **errfile**. Note the quotes. You need them if the spawned program has arguments of its own.

```
C:\> err trial errors
```

Run program **trial**, but send timing statistics to file **errors**.

If you wish to use **err** and **timer** to time **command.com**-internal commands, **err.c** must be changed to parse its arguments a little more precisely. Most people don't care how fast command.com clears the screen or does a DIR. I felt that you would be more interested in timing your own programs, so I left that simple exercise up to you. Once you have completed the performance tuning of your software (smirk) "piece of resistance," you can remove all the timing probes and have your program run at top speed by recompiling with the manifest constant, DEBUG, undefined. So either you

```
#define   DEBUG
```

or you:

```
#undef    DEBUG
```

to turn on timing or turn off timing, respectively.

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

```
C:\> test
figure.1        gettix.obj      listing.5       test.c          timer.obj
gettime.asm     gfile           oldtimer.c      test.exe        timer1.c
gettime.c       listing.1       oldtimer.exe    test.obj
gettime.obj     listing.2       t.c             timer.c
gettix.c        listing.3       t.exe           timer.exe
gettix.exe      listing.4       t.obj           timer.h
ls: 0.384478 secs.
value assign.: 0.000000 secs.
hello world again
hello world again
hello world again
hello world again
hello world again
hello world again
hello world again
hello world again
hello world again
hello world again
hello world again
hello world again
hello world again
hello world again
puts(): .109851 secs
XXXX
```

Code from "Time Is on My Side..." by Frank Greco

1

Listing 1

```
/*
 * gettime.c  --    Uses BIOS interrupt 0x1A to get the timer count
 *                  in clock ticks.
 */

#include <dos.h>

#define     TOD_INT     0x1a

unsigned long gettime()
{
    union REGS in, out;
    unsigned long current_tix;      /* 32-bit tick value to be returned */

    in.h.ah = 0;                    /* AH = 0    GET clock ticks */
    int86(TOD_INT, &in, &out);      /* Call BIOS timer tick interrupt */

    current_tix = ( (long) out.x.cx << 16 ) + out.x.dx;

/* If midnite passes, add a day's worth of clock tix to the return value */

    return ( out.x.ax & 0xff ) ? current_tix += 0x01800b0L: current_tix;
}
```

Listing 2

```
/*
 * gettix.c  -- Gets the current timer tick from the BIOS in two ways:
 *
 *          1.) Through the BIOS            - The preferred way
 *          2.) Directly via a far pointer  - The unrecommended way
 */

#include <dos.h>

main()
{
    unsigned long gettime();
    long far *fp;

    FP_SEG(fp) = 0x0040;
    FP_OFF(fp) = 0x006C;

    printf("BIOS: %lu   Thru far ptr: %lu\n", gettime(), *fp);
}
```

Listing 3

```
#define DEBUG                   /* Turn on timing statistics */

#define TIX_PER_SEC         18.20648            /* For the PC */
#define START()             _t1 = gettime()
#define STOP()              _t2 = gettime()
#define TIX_DIFF            (_t2 - _t1)
#define PRINT_TIME(cp)      fprintf(stderr,"%s: %lf secs.\n",\
                                cp, TIX_DIFF/TIX_PER_SEC)

#ifdef DEBUG
#   define TIME(code, string)   START(); code; STOP(); PRINT_TIME(string)
#else
#   define TIME(code, string)   code
#endif


static long _t1, _t2;           /* Tick values for calc. elapsed time */
```

Listing 4

```
/*
 * timer    -----   Gives a rough execution time of a particular process
 *
 * Uses:
 *
 *      timer.c     -   Picks off cmd line and assembles it for spawnvp()
 *      gettime.c   -   Pulls clock ticks from MS-DOS
 *
 */

#include <stdio.h>
#include <process.h>
#include <dos.h>
```

2

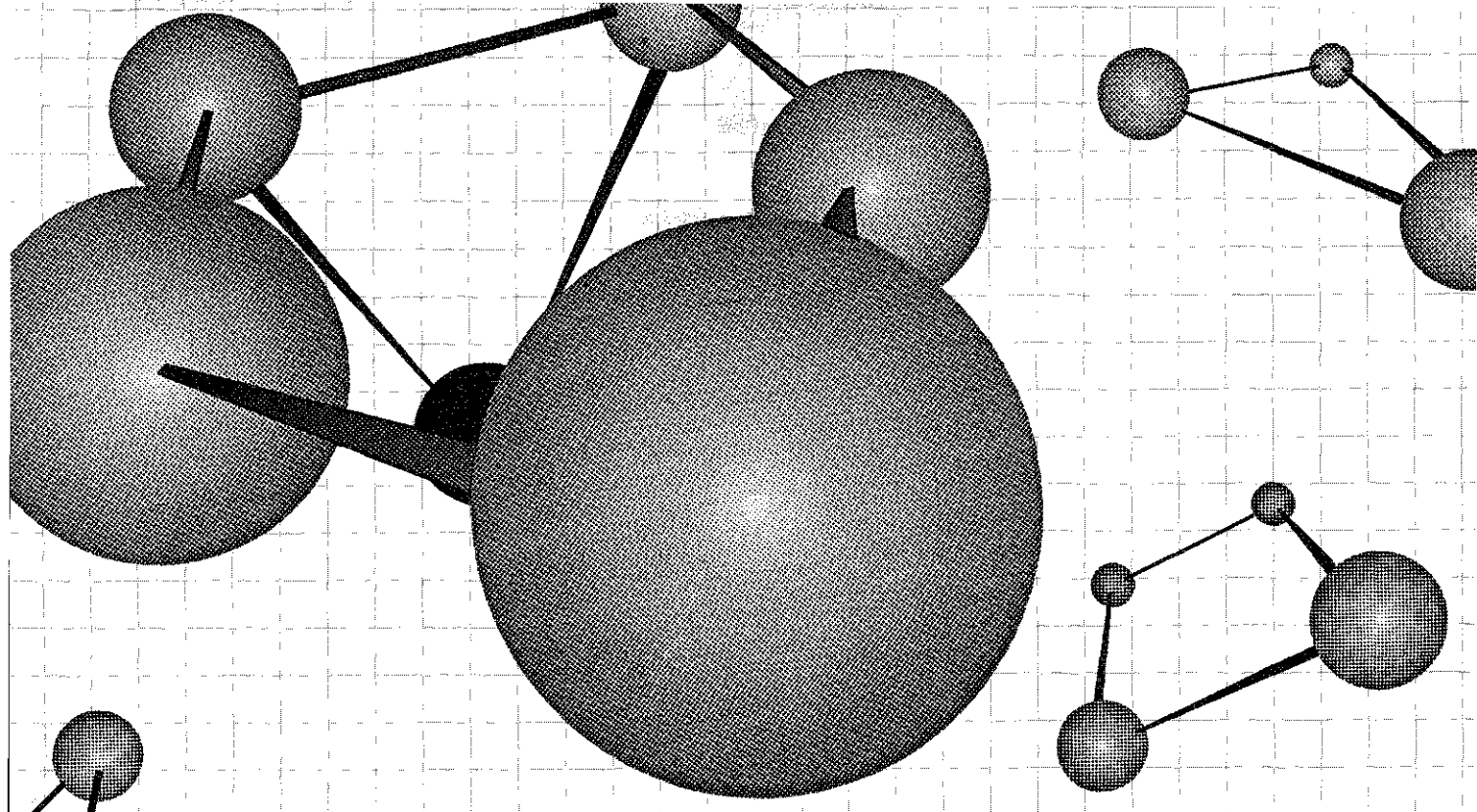
```
#include <errno.h>
#include "timer.h"

#define NO_ERR      0       /* No error to return to the shell */
#define NO_ARGS     1       /* If nothing to time, return this value */

extern int errno;

main(argc,argv)
char **argv;
{
    char *bp;                   /* Ptr to command to be spawned */
    unsigned long gettime();    /* Gets clock ticks from MS-DOS */
    int ret;                    /* To save the return value of spawn() */

    if (argc == 1)      /* If nothing to time, bye-bye */
        exit(NO_ARGS);

    bp = argv[1];       /* We need to avoid problems with side effects,
                         * so use another ptr to the command to be spawned.
                         */
/*
 * Now time the requested command
 */
    TIME(ret = spawnvp(P_WAIT, bp, ++argv), bp);

    if ( ret == -1 )        /* In case of an error with spawn */
        switch (errno)
        {
            case ENOENT:    fprintf(stderr, "File '%s' not found\n", bp);
                            exit(1);

            case ENOMEM:    fprintf(stderr, "%s to run '%s'\n",
                                "Not enough memory", bp);
                            exit(2);
/*
 *  Spawn() does not appear to recognize invalid executables properly,
 *  so it appears that the following error (i.e., ENOEXEC) will
 *  unfortunately never be caught.
 */
            case ENOEXEC:   fprintf(stderr, "Cannot execute '%s'\n", bp);
                            exit(3);

            default:        fprintf(stderr, "Cannot spawn '%s'\n", bp);
                            exit(4);
        }
    exit(NO_ERR);       /* Good Housekeeping */
}
```

Listing 5

```
#include <stdio.h>
#include <process.h>
#include <dos.h>
#include "timer.h"

main(argc,argv)
char **argv;
{
    char *bp;                   /* Ptr to command to be spawned */
    unsigned long gettime();    /* Gets clock ticks from MS-DOS */
    int value;                  /* Arbitrary variable */
    int i;                      /* Arbitrary variable */

    bp = argv[1];

    TIME(spawnvp(P_WAIT, bp, ++argv), bp);
    TIME( value = 1 + 3 * 8, "value assign.");

    START();
    for (i = 15; i--; )
        fputs("hello",stdout),fputs(" world", stdout),fputs(" again\n",stdout);
    STOP();
    PRINT_TIME("puts()");

    TIME(
    {   int i;
        for ( i = 0; i < 300; i++)
            putchar('X');
    }, "for loop"
        );

    exit(0);                    /* Good Housekeeping */
}
```

◆

PRODUCED 2005 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED



Aztec C - the *best* C in the Micro-cosm

Why is Manx Aztec C the best C ?

Because Manx offers a combination of performance, features, and pricing that are unbeatable.

What kind of performance ? In the 2/87 issue of Computer Languages, our 3.2 release of Aztec C for PC-DOS turned out 109,489 iterations of the Dhrystone benchmark per minute to108,303 for Microsoft C. Our 4.1 code is even faster.

What kind of features ? Features like in-line assembly code, a professional make utility, library source, ROM support, debuggers, assemblers, and overlay linkers. Many of these features are unavailable or extremely high priced options in other systems. Microsoft, for instance, does not provide library source or in-line assembly.

What kind of pricing ? For $199, the Professional version of Aztec C gives you an Optimized C, a macro assembler, a librarian, overlay linker, extensive UNIX and MS-DOS specific library routines, a powerful debugger, sample programs, and over 500 pages of documentation. Manx Aztec C has a hard earned and well deserved reputation as a leading C software development system for PC-DOS and MS-DOS. At this price, it is also one of the best software values available.

Over 80 percent of our sales are repeat sales or are based on recommendations. That says it all. At Manx, we extend ourselves to the limit to provide the best software, the best service, and the best support possible. Call today, and join the over 50,000 users who know the Manx difference.

Manx also provides C systems for Macintosh, Amiga, CP/M-80, TRS-80, Apple //, 6502/65C02 ROM, 8080/Z80 ROM, 680x0 ROM, and 8086/80x86 ROM. **Call for prices and information.**

Aztec C86 4.1

PC-DOS • MS-DOS • ROM

Aztec C86-p Professional System $199

• optimized C with near, far, huge, small, and large memory • in-line assembler • in-line 8087/80287 • ANSI support • Fast Float (32 bit) • optimization options • Manx Aztec 8086/80x86 macro assembler • Aztec overlay linker (large/small model) • source level debugger • object librarian • MS-DOS 3.x file sharing & locking • comprehensive libraries of UNIX, MS-DOS, Screen, Graphics, and special run time routines.

Aztec C86-d Developer System.....$299

• includes all of Aztec C86-p • Unix utilities make, diff,grep • vi editor • 6+ memory models • Profiler.

Aztec C86-c Commercial System.....$499

• includes all of Aztec C86-d • Source for library routines • ROM Support • CP/M-86 cross dev. option • One year of updates.

How To Become An Aztec C User

To become an Aztec C user call **800-221-0440.** From NJ or international locations call **201-542-2121. Telex: 4995812** or **FAX: 201-542-8386. C.O.D., VISA, Master Card, American Express, wire (domestic and international), and terms are available.**

Aztec Systems bought directly from Manx have a 30 day satisfaction guarantee. Systems are upgradable by paying the difference in price plus $10. One and two day delivery is available for all domestic and most international destinations.

MANX

Manx Software Systems
1 Industrial Way, Eatontown, NJ 07724

To order or for more information call today.

1-800-221-0440

In NJ or international call (201) 542-2121 • TELEX: 4995812

REPRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

Porting Mainframe Software to the PC

The Software Approach—FORTRAN Compilers under DOS

by
M. Steven Baker

Have you ever wanted the power of a minicomputer or dedicated workstation on your desktop? Are there large mainframe or minicomputer applications that would be much more convenient if they could run on your lowly PC in a reasonable time and not cost a fortune? Are you tired of waiting for the turnaround on jobs using a remote VAX or IBM mainframe?

This article and the its companion on the Definicon coprocessing engines are about porting larger mainframe and minicomputer programs to the PC environment. I will focus here on porting FORTRAN programs using DOS FORTRAN compilers, but many of these remarks apply to moving programs written in other languages as well. The emphasis of this review of DOS FORTRAN compilers is on the porting task, not on whether a compiler would be good as an educational tool or best implements the latest FORTRAN standard.

FORTRAN (FORmula TRANslator) is one of the oldest computer languages and lacks many of the features available in newer languages such as Pascal, Modula 2, Ada, and C. Compilers for the FORTRAN language are more difficult to write than those for C or Pascal, but FORTRAN continues to be a mainstay of the scientific and engineering fields and is widely used on larger computers. Many scientific subroutine and other supporting packages are available to mainframe and minicomputer FORTRAN programmers.

The power of FORTRAN is in fast number crunching, and of all the programming languages, FORTRAN is the one most likely to be supported across mainframe and minicomputer environments. Traditionally, FORTRAN programs have been the most portable across these environments. (Note: I'll use the word "mainframe" to include minicomputers in the following discussion).

Even so, a few years ago, any FORTRAN program (i.e., statistics) ported to DOS was almost guaranteed to be bug ridden because of bugs in the compilers available. The ancient roots of the FORTRAN language are still evident in the latest adopted standards for the language. In FORTRAN, keywords can also be used for variable names, and spaces are not delimiters. This makes language translation a difficult task as evidenced by the limited supply of DOS FORTRAN compilers.

Stumbling Blocks

What are the problems involved in squeezing mainframe code into the PC? The first is inherent in the memory limits of the 8086 CPU and DOS. The 8086 supports up to 1 megabyte of memory. The PC hardware environment and MS-DOS further limit available memory to 640K. DOS itself takes less than 40K, leaving about 600K for user programs. Mainframe and minicomputer software runs in virtual memory. If I need a large FORTRAN array of real numbers on a mainframe, I just define a big common block, maybe AA(100000). Any sorting to be done just uses a huge array. This is a perspective vastly different from a microcomputer programmer's or even that of early Unix programmers operating in a small memory environment. There might have been other ways to design the program, but why bother on a mainframe?

The Linpack benchmark program from Argonne National Laboratory does matrix inversion on several arrays of reals AA(200,200)—that's 156K each. But testing the benchmark with double precision (eight-byte reals) just can't be done on the PC with an 8086/88.

Another problem is those #@& 64K memory segment limits of the 8086 and 80286 CPU. I'm not aware of any language compilers under DOS that support single procedures or functions (Program Units in FORTRAN jargon) larger than a single 64K segment. Again, mainframe subroutines don't have these limits. The other impact of segments is the inherent speed penalty when you access data that may spill over a single segment. In FORTRAN, common blocks and arrays can't be limited to a single 64K data segment. If they can't exceed 64K, porting would be virtually impossible.

The final difficulty lies in the FORTRAN language itself, in the variety of its language extensions, and in the forgiving quality of mainframe (VAX and IBM) FORTRAN compilers. The FORTRAN language was standardized in 1966 (ANSI FORTRAN 66) and then again in 1978 (ANSI FORTRAN-77). The ANSI 1977 standard defines two implementations: a FULL implementation and a SUBSET implementation. Partly due to limitations in the 1966 standard, most mainframe FORTRAN compilers supported a large set of extensions to the 1966 definition. ANSI FORTRAN 77 included some of the common extensions, but left a number out. Many, but unfortunately not all, of these mainframe extensions have been carried over to the DOS FORTRAN compilers.

Probably more important, mainframe compilers (VAX and IBM) are very forgiving of violations of the standard. What's truly amazing is that now, almost ten years from the adoption by ANSI of FORTRAN 77, mainframe and DOS compilers must still support features from the 1966 standard of 20 years ago in order to be functional today. Recently written mainframe FORTRAN software may still depend on features that are not included in FORTRAN 77. A new FORTRAN 8X standard has been under slow development for the last several years.

DOS FORTRAN Compilers

I tested eight different FORTRAN compilers that run under DOS with some standard benchmark programs and on source code originally written for a mainframe. Table 1 gives an overview of the compilers reviewed. Most are newer versions of DOS compilers that have been available for some time. During the review period, I received revised software from five of the vendors. The Intel FORTRAN compiler was originally designed to operate under Intel RMX-86, but was later ported to the DOS environment. The PECAN FORTRAN 77 Compiler runs under the UCSD P-System under DOS, but can generate stand alone programs. Lahey actually offers three FORTRAN compiler

PRODUCED 2005 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

products: (1) F77L—their full compiler, (2) F77L-EM for use on 80286/80386 machines with extended memory, and (3) LP-77 Personal FORTRAN77 compiler—a subset of F77L. These DOS compilers range in price from $99 to $750. The Intel compiler needs the Intel math Libraries, Linker, Assembler, and DOS interface software, which also costs $750.

For a substantial fee, the National Bureau of Standards certifies FORTRAN and other language compilers to have successfully executed a validation suite of programs. The certification lasts for one year. After this time, a compiler vendor must re-apply and either assert that no substantive changes have been made or go through the validation procedure anew. The FORTRAN Validation Suite consists of just under 200 test programs. For those brave souls wishing to review this validation suite or test their current compilers, the mag tape with the test suite costs a mere $2800. The substantial testing cost is an additional expense. Microsoft FORTRAN 4.0, IBM Professional FORTRAN 1.00 and 1.01, Ryan-McFarland 2.11 (now Austec), and Prospero FORTRAN Version 1.15 were the only DOS compilers validated at the FULL level on the NBS certification list dated July 1987. The IBM FORTRAN compiler is an older version of the Ryan-McFarland product. Microsoft FORTRAN Version 3.31 was certified as a SUBSET compiler to February 1988. Digital Research's DR FORTRAN-77 Version 4.0 was dropped from the certified list on August 1985, probably because DRI did not re-apply. AT&T FORTRAN 77 running under Unix V was also dropped from the previous certified list. My experience has been that NBS certification is no guarantee that a compiler is best for porting mainframe software.

With the exception of Pecan's and Intel's, these compilers all implement the FULL ANSI FORTRAN 77 language. The Intel compiler supports the SUBSET definition of the ANSI FORTRAN 77 standard, but does not omit much from the FULL standard. The Lahey Personal FORTRAN 77 (LP-77) complies with the FULL ANSI standard, but it's a subset of their main Lahey compiler F77L. LP-77 limits the total FORTRAN code and data space to 64K each. For smaller mainframe programs, this may be adequate, but most programs will require larger space.

The Pecan compiler does *not* implement even the SUBSET language definition. This UCSD compiler has one fatal omission that makes it virtually useless for porting software. Integers in the Pecan FORTRAN-77 system can be only two bytes long, an obvious holdover from the Pascal system on which it rides. In FORTRAN, real, integer, and logical variables must be able to take the same space (four bytes) because older programs often equivalenced them. This also limits array sizes to 32,767 elements, another unrealistic assumption for mainframe programs. Because of these limitations, this compiler will not be discussed further.

Table1. DOS FORTRAN Compilers Reviewed

Product	Austec RM/FORTRAN	DRI FORTRAN-77	Intel FORTRAN-86	Lahey F77L	Microsoft FORTRAN	Pecan FORTRAN-77	Prospero ProFOR-77	Watcom WATFOR-77
Version	2.42	4.1	3.0	2.21	4.01	IV.2.2	1.232	2.0
Price	$595	$350	$750	$477	$450	$99.95	$390	$375
Disks (360K)	4	2	1	2	7	3	3	4
ANSI Standard	Full	Full	Subset	Full	Full	----	Full	Full
NBS Certified	Yes	Dropped	No	No	Yes	No	Yes	No
Source Debugger	Yes	No	Yes	Yes	Yes	No	Yes	Yes

Compiler Features

Table 2 is the compiler features table mandatory in any review. These compilers support a variety of command line and source code options. ANSI FORTRAN 77 contains several appendices that are not part of the official standard document. The features described in these ANSI appendices are extensions"condoned" by ANSI. Extensions make up an unfortunately large part of the features that must be supported by DOS and mainframe compilers. For comparative purposes, I have included VAX FORTRAN Version 4.0 and IBM VS FORTRAN 4.1. The bottom line is that no one yet sells the equivalent of VAX or IBM FORTRAN compilers for the PC.

Real Machines Eat Math Chips

From a mainframe perspective, many features of these compilers are irrelevant. FORTRAN is a language for math intensive stuff. Emulation libraries for the math chip do number crunching at a snail's pace compared to the real thing (80x87). Compiler options supporting the small and medium models (i.e., 64K limits for code or data) may be okay for educational exercises but not for large problems. Libraries supporting older versions of the compiler (a.k.a. Microsoft) are also of limited value. In many cases, the debuggers may also be of very limited value. If the debugger takes significant space, this may prevent its use with mainframe source

Table 2. Compiler Features and Extensions

								MAINFRAME	
	Austec	DRI	Intel	Lahey	Microsoft	Prospero	WATCOM	VAX	IBM
Standard Types:									
INTEGER(bytes)	2/4	2/4	1/2/4	2/4	2/4	4	4	2/4	4
REAL	4	4	4	4	4	4	4	4	4/8
DOUBLE PRECISION	8	8	8	8	8	8	8	8	8/16
LOGICAL	1/4	1	1/4	1/4	2/4	4	4	2/4	4
COMPLEX	8	8	8	8	8	8	8	8	8/16
CHAR* (max)	255	65,535	65,535	65,280	32,767	32,767	65,535	65,535	32,767
Extension Types:									
INTEGER*n	2,4	1,2,4,8	1,2,4	2,4	1,2,4	1,2,4	1,2,4	2,4	2,4
REAL*n	4,8	4,8,10	4,8,10	4,8	4,8	4,8	4,8	4,8,16	4,8,16
LOGICAL*n	1,4	None	1,2,4	2,4	1,2,4	1,2,4	1,4	1,2,4	1,4
COMPLEX*n	8,16	8,16,20	8,16	8,16	8,16	8	8,16	8,16	8,16,32
BYTE	-	-	-	-	-	-	-	1	-
ANSI Extensions:									
Hollerith DATA	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
FORTRAN-66 DO	Yes	Yes	Yes	No	Yes	No	No	Yes	Yes
ENCODE/DECODE	No	Yes	No	No	No	No	No	Yes	Yes
NAMELIST	No	No	No	Yes	No	No	No	Yes	Yes
AND numbers	No	No	Yes	No	No	No	No	Yes	No
Recursion	No	No	Yes	Yes	No	No	No	No	No
Source Code:									
Flag non-ANSI	Yes	No	No	No	Yes	No	Yes	Yes	
Free format	Yes	Yes	Yes	Yes	Yes	No	No	Yes	Yes
Inline comment	No	No	No	Yes	Yes	No	No	Yes	No
INCLUDE	Yes	Yes	Yes	Yes	Non-stand	Yes	Non-stand	Yes	
Vax-like DEBUG	Yes	No	No	No	Yes	No	No	Yes	No
Variable names:									
Global (max)	8	40	31	31	31	6	32	31	6
Local (max)	31	40	31	31	31	6	32	31	6
Special symbols	$	$_	_	$_	$		$	$_	$
Intrinsic Routines:									
Bitwise	Yes	No	No	Yes	Yes	No	No	Yes	Yes
Command-line	Yes	No	No	Yes	Yes	No	No	Yes	No
Date and Time	Yes	No	No	Yes	Yes	Yes	Yes	Yes	Yes
Mixed Languages	Yes	No	Yes	Yes	Yes	No	No	Yes	No

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

Fortran Support for IBM PC/XT/AT & Compatibles

Versions Available For:
Microsoft, Supersoft, RyanMcFarland, IBM Professional, Lahey, & IBM Fortran.

Forlib-Plus
$69.95

Supports graphics, interrupt driven communication, program chaining, and file handling/ disk support. A Fortran coded subroutine is included which will plot data on the screen either in linear/linear, log/linear, linear/log, or log/log on the appropriate grid.

Strings & Things
$69.95

Supports string maipulations, command line usage, DOS call capabilities, SHELL generation and data transmission, BATCH file control, music generation, PEEKS and POKES, PORT access, and general register manipulations.

For-Winds
$89.95

Gives the Fortran programmer the capability of generating up to 255 windows on the screen. Each window can be individually scrolled, moved, sized, generated, and removed. Both color and monochrome type displays are supported. Full source code is supplied for customization.

ACS Time Series
$495.00

This is a COMPLETE time series analysis package which contains VERY HIGH SPEED FFTs, Filter generations, convolutions, transfer function calculations, auto and cross spectra calculations, Cepstrum, curve fitting algorithims, coherence calculations, and many other associated routines. The price includes FULL source code.

Fortran Scientific Subroutine Package
$295.00

There are approximately 100 Fortran subroutines included which fall under the following 12 categories:
1) Matrix storage and Operations 2) Correlation and Regression, 3) Design Analysis (ANOVA), 4) Descriminant Analysis, 5) Factor Analysis, 6) Eigen Analysis, 7) Time Series, 8) Nonparametric Statistics, 9) Distribution Functions, 10) Linear Analysis, 11) Polynomial Solutions, 12) Data Screening. Full source code is included.

ALPHA COMPUTER SERVICE
5300 ORANGE AVENUE SUITE 108
CYPRESS, CALIFORNIA 90630
(714) 828- 0286

California Residents Include 6% Sales Tax

There are NO license fees

Number **152** on the Reader Service Card

programs (i.e., 200K+ overhead of Codeview). Finally, assembly language output for bit fiddling is unlikely with 5,000 line source programs.

Austec (formerly Ryan-McFarland)
Austec recently purchased Ryan-McFarland and now sells their FORTRAN and COBOL products. I reviewed RM/FORTRAN 2.40, their current version released in August 1987. Older versions of RM/FORTRAN are sold by IBM as IBM Professional FORTRAN, based on RM/FORTRAN 1.1 from 1985. IBM is soon to release FORTRAN/2 based on RM/FORTRAN 2.1 for OS/2. RM/FORTRAN is also available under Xenix. Technical support is available, but preference is given to distributors and dealers, not end users.

The latest RM product has added a FORTRAN development environment called FORTE with an integrated editor, syntax checker, and project make features. The compiler is bundled with an overlay linker and librarian from Phoenix (subset of the full PLINK86+) and an EXE file compression utility (equivalent to the DOS LINK with /E switch). RM/FORTRAN comes with both 5.25- and 3.5-inch disks.

RM/FORTRAN continues as a solid but conservative FORTRAN compiler. The compiler is written in assembly language, but compiles quite slowly. The code generated for number crunching is some of the fastest available from any of the compilers tested. Version 2.40 was released with a number of embarrassing bugs. For instance, the debugger would lock the machine when the tutorial was run. A fix disk was later issued to correct problems. This latest version is 2.42. The compiler performs optimizations that may be disabled. It creates statics (SAVEd) for all variables and supports large arrays and common blocks. The compiler can generate both listing and cross reference reports. It can also emit a pseudo assembly language listing to stdout, which may be redirected to a file. The source code debugger requires a complete recompile and relink in order to be used.

I did not use the FORTE environment for this review. Like other programmers, I already have a fast editor I prefer to use. I was also concerned about using memory that the compiler and linker could better take advantage of. Microsoft LINK 3.05 was used for a linker to make comparisons more equal with the other compilers and not penalize RM link times by using PLINK86.

The documentation is divided into three manuals (*User's Guide*, *FORTE*, and *Language Reference*) plus several smaller sections on installation and the Phoenix linker and librarian. Each main manual has a separate table of contents and index. The *User's Guide* is specific to the PC environment while the *Reference Manual* applies to both DOS and Xenix. They're quite good, but do not indicate which features are part of the ANSI 77 standard and which are extensions.

With mainframe code, be forewarned—use the /Z1 switch (store into unSAVEd local variables from registers at subroutine exit). RM/FORTRAN can generate 80286/80287 instructions for a small improvement in size and speed. The compiler is based on a number of fixed size tables for expressions, do loops, etc., documented in the *User's Guide*. If these table sizes are exceeded, undocumented errors—such as Compiler Error, Number 8002: Roll 3 Phase 34 (translation: we crashed in the optimization phase expression analyzer)—are reported. Rather than require a phone call for support, it would make much more sense to have a listing of what these error numbers mean for the user. RM has trouble with larger procedures even with optimization turned completely off. The fixed size tables are the culprit. The file I/O library appears to support large record sizes but is slow in speed.

Digital Research
DRI dropped further development of its language products in 1985. Version 4.1 of their FORTRAN-77 compiler was the last revision. It is now marketed by another firm. I discovered in my discussions with DRI support personnel that FORTRAN-77 is considered a "mature" product (no product support is necessary or available). The marketing firm offers only limited user technical support.

FORTRAN-77 was written in C and compiles at a moderate speed. The package comes with a DRI Linker (LINK86) and Librarian (LIB86). I had to write my own time routines in assembly for benchmarking. The linker does not work reliably with object files generated by MASM. I had to use DRI's assembler RASM86, which is part of DR Assembler+Tools and costs an additional $200. I did find RASM86 to be faster than MASM and easy to use, but RASM86 does not support the same assembler directive syntax that MASM does. The compiler can generate both listing and cross reference reports. It can also emit an intermixed source code/assembly language listing. No debugger is included. The DR Assembler+Tools includes an assembly language debugger, SID86, which may be used with LINK86 output, but there is no FORTRAN source code debugger. The documentation has a good table of contents and index.

The compiler generates slower numeric code than most of its rivals. The manual states that each large array must be in a separately named COMMON segment. The Linpack benchmark, which uses several large arrays, crashed during execution. The compiler appears to be running as a small model program (64K code and 64K data), which severely limits the size of source code that can be compiled. DR FORTRAN couldn't swallow large files or files with large program units. It would just die without giving meaningful errors. FORTRAN-77 error messages are not listed in the manual. They are supposed to be mailed to you upon receipt of the Registration Card.

Continued . . .

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

UNLEASH YOUR 80386!

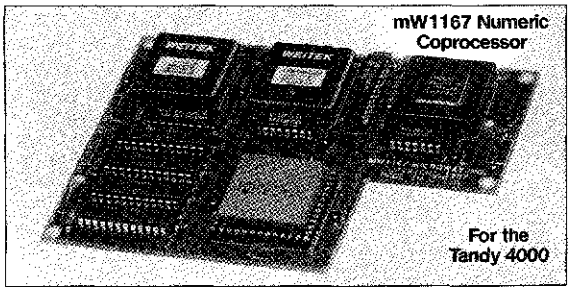
Your 80386-based PC runs at least twice as fast as your old AT. This is good, but not great. The products described below will unleash the true potential of your 80386, giving you 4 to 16 times the power of your old AT. These new MicroWay products include a family of 80386 native code compilers and the mW1167 numeric coprocessor.

Examples of the increases in capacity and performance include:

- Programs compiled with MicroWay NDP Fortran-386 execute 2 to 8 times faster than those compiled with existing 16-bit Fortrans. NDP Fortran-386 can also address up to 4 gigabytes of memory instead of the standard 640 kbytes. MicroWay's NDP compilers and the programs they generate run on MS-DOS or Unix V.
- NDP Fortran-386 generates code for the 80287, 80387 or MicroWay's mW1167. The mW1167 has a floating point throughput exceeding 2.5 megaflops, which is 4 to 5 times the throughput of an 80387 and is comparable to the speed achieved by the VAX 8600.

Equally important, whichever MicroWay product you choose, you can be assured of the same excellent pre- and post-sales support that has made MicroWay the world leader in PC numerics and high performance PC upgrades. For more information, please call the Technical Support Department at

617-746-7341



MicroWay® 80386 Support

MicroWay 80386 Compilers

NDP Fortran-386 and **NDP C-386** are globally optimizing 80386 native code compilers that support a number of Numeric Data Processors, including the 80287, 80387 and mW1167. They generate mainframe quality optimized code and are syntactically and operationally compatible to the Berkeley 4.2 Unix f77 and PCC compilers. MS-DOS specific extensions have been added where necessary to make it easy to port programs written with Microsoft C or Fortran and R/M Fortran.

The compilers are presently available in two formats: Microport Unix 5.3 or MS-DOS as extended by the Phar Lap Tools. MicroWay will port them to other 80386 operating systems such as OS/2 as the need arises and as 80386 versions become available.

The key to addressing more than 640 kbytes is the use of 32-bit integers to address arrays. NDP Fortran-386 generates 32-bit code which executes 3 to 8 times faster than the current generation of 16-bit compilers. There are three elements each of which contributes a factor of 2 to this speed increase: very efficient use of 80386 registers to store 32-bit entities, the use of inline 32-bit arithmetic instead of library calls, and a doubling in the effective utilization of the system data bus.

An example of the benefit of excellent code is a 32-bit matrix multiply. In this benchmark an NDP Fortran-386 program is run against the same program compiled with a 16-bit Fortran. Both programs were run on the same 80386 system. However, the 32-bit code ran 7.5 times faster than the 16-bit code, and 58.5 times faster than the 16-bit code executing on an IBM PC.

NDP FORTRAN-386™ **$595**
NDP C-386™ **$595**

MicroWay Numerics

The **mW1167™** is a MicroWay designed high speed numeric coprocessor that works with the 80386. It plugs into a 121 pin "Weitek" socket that is actually a super set of the 80387. This socket is available on a number of motherboards and accelerators including the AT&T 6386, Tandy 4000 and MicroWay Number Smasher 386 (Jan. '88). It combines the 64-bit Weitek 1163/64 floating point multiplier/adder with a Weitek/Intel designed "glue chip". The mW1167™ runs at 3.6 MegaWhetstones (compiled with NDP Fortran-386) which is a factor of 16 faster than an AT and 3 to 5 times faster than an 80387 .....**$1495**

Monoputer™ - The INMOS T800-20 Transputer is a 32-bit computer on a chip that features a built-in floating point coprocessor. The T800 can be used to build arbitrarily large parallel processing machines. The Monoputer comes with either the 20 MHz T800 or the T414 (a T800 without the NDP) and includes 2 megabytes of processor memory. Four or more Transputers can be easily linked together to form a Quadputer. A single T800 is comparable in speed with an mW1167-equipped 80386. The compilers to drive one or more Monoputers include Occam, C, Fortran, Pascal and Prolog.

Monoputer T414-20[1] **$1495**
Monoputer T800-20[1] **$1995**
Biputer™ T800/T414[2].....**$4995**
Quadputer™ T414-20[2] **$6995**

[1]Includes Occam [2]Includes TDS

80287 ACCELERATORS

287Turbo-10 .....**$450**
287Turbo-12 .....**$550**
287TurboPlus-12 .....**$629**

80386 Multi-User Solutions

AT8™ - This intelligent serial controller is designed to handle 8 users (16 with two boards) in a Xenix or Unix environment with as little as 3% degradation in speed. It has been tested and approved by Compaq, Intel, NCR, Zenith, and the Department of Defense for use in high performance 80286 and 80386 Xenix or Unix based multi-user systems **$1299**

MicroPort Unix 5.3 is a port of the new Unix 5.3 to the 80386. MicroWay NDP-386 compilers currently run on this version of UNIX.
MicroPort Unix 5.3 .....**from $399**

PC-MOS-386™ is an 80386 operating environment that turns an AT with an AT8 into an MS-DOS multi-user system. The system makes it possible to run applications such as Lotus 1-2-3 on terminals. The operating system also has a Phar Lap compatibility mode that runs programs developed with the Phar Lap versions of MicroWay's compilers .....**from $199**

Phar Lap™ created the first tools that make it possible to develop 80386 applications which run under MS-DOS yet take advantage of the full power of the 80386. These include an 80386 monitor/loader that runs the 80386 in protected linear address mode, an assembler, linker and debugger. These tools are required for the MS-DOS version of the MicroWay NDP Compilers.
Phar Lap Tools.....**$399**

MATH COPROCESSORS

80387-16 16 MHz.....**$495**
80287-10 10 MHz.....**$349**
80287-8 8 MHz **$259**
80287-6 6 MHz **$179**
8087-2 8 MHz **$154**
8087 5 MHz **$99**

MicroWay

The World Leader in PC Numerics

P.O. Box 79, Kingston, Mass. 02364 USA (617) 746-7341
32 High St., Kingston-Upon-Thames, U.K., 01-541-5466

Number **104** on the Reader Service Card

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

Q. How many programmers does it take to maintain a MAKE dependency file?

A. NONE! *If you use* VersiMAKE™

VersiMAKE™ is a full-featured MAKE utility that includes:

- **Dependency Generation**
  Derives your system's dependencies, through analysis of its C and MASM source files. Say goodbye to manual maintenance of MAKE dependency files!
- **Wild Card File Name Matching**
  Analyzes an entire collection of source files with a single statement.
- **Nested Include File**
  Handles standard C and MASM "include" conventions, and the INCLUDE environment variable.
- **Powerful Macro Facilities**
  Built-in macros, user-defined macros, and environment variables.
- **Analytical Reports**
  Shows the entire Include file hierarchy for each source file analyzed, and all of the parent source files for each Include file.

Q. How many programmers does it take to trace a symbol thru your system?

A. ONE! *If you use* VersiCREF™

VersiCREF™ is a unique utility that creates a Master Cross-Reference of your entire system.

- **Multi-Lingual**
  Handles C, assembler, or both.
- **Flexible**
  File names with line numbers, or file names alone. Global and local symbols, or globals alone.
- **Powerful**
  Easily handles systems containing 100 source files or more.

VersiMAKE™	$125
VersiCREF™	$75
Both	$150

(Outside U.S., add $5 for shipping and handling)

800-334-4096
(In NJ, 609-871-0202)
MC/VISA/AMEX accepted

SUMMIT INFORMATION SYSTEMS, INC.
73 East Lane, Willingboro, NJ 08046

Number **158** on the Reader Service Card

Intel

Intel's FORTRAN-86 is considered one of their DOS Development Tools. This compiler was originally written to run on Intel Development systems using a Universal Device Interface (UDI). A UDI2DOS (UDI to DOS) converter was attached to allow execution under DOS. This is officially a SUBSET compiler; however, only a few features from the full ANSI standad are omitted. The most significant omissions are the INQUIRE statement, ENTRY and alternate returns, and several FORMAT edit descriptors (T, TR, TL, S, SS, SP, Iw.m). FORTRAN-86 also has one very inconvenient feature. The source line size is not limited to 72 characters, so any sequence numbers past column 72 are interpreted as bad source line statements. Unfortunately, there is no way to turn this "feature" off. Source files must also be stripped of the Control-Z characters appended by many editors.

The compile times are slower than most of the other compilers'. Linking is a multi-step process. After linking with Intel's linker, the output file must be run through a UDI2DOS utility to patch the DOS interface onto the output file and make a DOS executable image. For benchmarks, I have considered this step part of the link time. The link step is also long since the supplied FORTRAN runtime library is spread across about eight files. If you used FORTRAN-86 extensively, it would be best to combine these library files into one or two larger libraries and try to reduce link time. FORTRAN-86 can generate both listing and cross reference reports and can also generate an assembly language source file. A source language debugger, PSCOPE86, is included with the Development Tools package. A problem for the user is the reporting of compiler errors. Errors are reported by the sequence number of the executable statement from the beginning of the Program Unit rather than by the line number in the file. Most often, in order to locate an error, a complete listing file must be generated and searched. In order to use the debugger, a complete re-compile and re-link is required.

In Intel's's documentation, a shaded background is inconsistently used to identify some differences or extensions from ANSI FORTRAN 77, but this feature was not documented anywhere in the manual. FORTRAN-86 was the most difficult compiler in setting up files for the linker. The documentation on the disks was not up to date. Files required for linking were not clearly noted or used in examples. It took me several hours of trial and error to puzzle out the required library files needed for 8087 inline code.

The code generated by this compiler is average in speed. It defaults to INTEGER*2 and LOGICAL*1 variables. The compiler was not able to compile the Linpack benchmark because of the complexity of one expression—about 16 levels of parentheses; however, this would have been easy to fix by breaking the one offending statement into two. FORTRAN-86 was one of only two to compile and execute the Superlite benchmark program. Apparently, it can swallow large program units.

Balanced against all its weaknesses, this compiler and linker has one outstanding feature not duplicated in any of the other products. The compiler incorporates TYPDEF records into the object file so that the linker actually does variable and argument type checking between subroutines during the link process. This linkage check of subroutines and variables can identify source code errors even when the modules are compiled separately. For example, it found array dimension errors in the CHILC program from one file to another. Unfortunately, the DOS linkers don't support this powerful feature.

Lahey

Lahey F77L was by far the easiest compiler to use. With the exception of WATFOR, F77L compiled much faster than the other products. Compile and link times were typically four to five times faster than the Austec or Microsoft products'. The compiler has several outstanding features that ease its use. On startup, the compiler looks for a file F77L.FIG containing the default command line options. Any actual command line options will override the parameters in this file. This makes it easier to work on a project—just create the appropriate F77L.FIG file. The compiler can also be instructed to display selected options at compile time and in the listing file. A second feature is the F77L.FIX file. On startup, F77L looks for this file, which contains fixups to the compiler to correct bugs discovered after initial distribution. The compiler self-patches itself and then executes. Quite nice, I'd say. This allows a user to dial into the Lahey Bulletin Board (provided as part of their technical support) and download the latest bug fixes as F77L.FIX (a very small file) and just keep working.

F77L supports recursion but also has a command line option to SAVE all local variables as statics. The compiler can generate both listing and cross reference reports. Compiler diagnostics, however, were good enough that I exercised this feature only once for testing purposes. Other options allow bounds checking of array subscripts, checking of subroutine interfaces, and protecting numeric constants as arguments to subroutines at runtime. Statement line numbers may also be included in the object file for a complete traceback at runtime error. Bounds checking may have to be turned off with mainframe programs that don't bother to properly define array dimensions in subroutines. The compiler can generate output for Lahey's FORTRAN source code debugger (SOLD) without requiring a recompile or relink. The debugger data is output to an SLD file, not included in the object file. The compiler diagnostics warn of unused variables and catch argument errors when the number of arguments to a subroutine or function differ from its definition. The compiler does not, however, catch argument type mismatch errors.

Continued . . .

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

o new o written specifically for Turbo C o compatible with the PC, AT, & PS/2 o convert a Turbo C program to a TSR activated by a hotkey o create screens, windows, & menus with an interactive designer o rapidly display screens & strings o read data from specific fields on the screen o display, stack, & move windows o control the cursor, video modes, & display attributes o write data to a large virtual screen o scroll & view a virtual screen through a window o manage the keyboard o manage the printer o swap printer ports o format printed reports & forms o access the DOS print queue from Turbo C o manage directory names, volume labels, & file dates & times o open up to 255 file handles simultaneously o control-break & critical-error handlers o a complete device independent graphics package o an LIM expanded memory interface o many more functions & utilities o 2 diskette set o ready-to-link library files for all memory models o complete commented source code included o printed manual o $39.50 introductory price o

Systems Tools for Turbo C

A Comprehensive Function Library by the Author of *Systems Programming in Turbo C* (SYBEX,1988)

Send check or money order for $39.50 plus $2.50 shipping & handling (CA residents please add 6% sales tax). Your Turbo C toolset will be shipped immediately.

Michael J. Young, Software Engineering
P.O. Box 5068
Mill Valley, CA 94942
415/383-5354

Number 168 on the Reader Service Card

PLOT TEXT ON ANY GRAPHICS SCREEN!!

YES, we said ANY graphics screen, even VGA! ***Finally!*** **Xgraf** is a super set of low level assembly graphics routines that you call directly from Compiled Basic. The **Xgraf** kernel replaces Basic's confusing statements with consistent full featured calls specifically designed for the Basic programmer. **Xgraf supports Hercules, CGA, EGA, EEGA, VGA, and user defined screen sizes.** All modes can be accessed in real time - no recompiling!

Advanced features extend Basic graphic capabilities!

In addition to Basic's graphics features, **Xgraf's *advanced features*** include zooming, tiling, a virtual screen page, screen compression to disk, windowing, screen importing, *and much more!*

***Finally!* Xgraf is only $99.00 + $4.00 s&h**
(Xgraf requires the Microsoft QuickBASIC compiler)

We *specialize* in libraries and tools for Compiled Basic. Our catalog features the ***Finally!*** family of programmer's libraries and *other top flight tools.*

KOMPUTERWERK - YOUR BASIC RESOURCE

KOMPUTERWERK
851 Parkview Blvd.
Pittsburgh, PA 15215

Call: 1-800-423-3400
PA & AK call (412) 782-0384
VISA and MC accepted!

Number 155 on the Reader Service Card

LAHEY SETS NEW FORTRAN STANDARDS

LAHEY PERSONAL FORTRAN 77 - The full ANSI 77 Standard and Debugger for $95.00.

F77L - "EDITOR'S CHOICE" *PC Magazine*
"...the most robust compiler tested." *Micro/Systems Journal*
"...the ultimate fully-blown ANSI FORTRAN 77... a fantastic product." *PC Australia*

F77L-EM - Breaks the DOS 640K program barrier with extended memory. The most powerful PC FORTRAN available.

FORTRAN TOOLS: Profiler, Mathematical Functions Library, Overlay Linker, Utility Libraries, Widows Library, Toolkit.

The difficult decision is not from whom you should buy your PC FORTRAN, but which Lahey compiler you should purchase.

Call us today to discuss your PC FORTRAN needs.

FOR INFORMATION OR TO ORDER:
1-800-548-4778
30-Day Money-Back Satisfaction Guarantee
Lahey Computer Systems, Inc., Box 6091, Incline Village, NV 89450 Tel: 702-831-2500 TLX: 9102401256

Lahey Computer Systems Inc.

Number 44 on the Reader Service Card

MICROCACHE™ and SPEEDPRINT™

The hottest PC utilities you can get.
Or your money back!

MICROCACHE™ monitors disk use and "learns" to anticipate your data accesses. Then, by satisfying these requests directly from RAM instead of disk, your programs run as if they are SUPERCHARGED.

- 100% Transparent Operation
- Uses any spare RAM available
- EMS/EEMS memory support
- SPEEDPRINT™ buffer option
- Requires NO program changes
- For IBM and compatibles

Introductory Price $89.00

MasterCard VISA

Az-Tech Software, Inc.
305 East Franklin
Richmond, MO 64085
(800) 227-0644 . . . (816) 776-2700

Number 51 on the Reader Service Card

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

The documentation in a PC binder includes a good table of contents and index. Extensions to ANSI FORTRAN 77 are differentiated by a shaded background. The Lahey compiler is the only product that requires a math coprocessor for execution, but this is of no consequence for large applications where speed is of utmost importance. The compiler does not output an assembly language listing.

The code generated for number crunching programs was above average in speed. F77L was the only compiler consistently able to swallow large procedures and files, and F77L was the only product able to compile and execute properly all of the test programs. SOLD, the source code debugger, was small enough (15K) and kept its help and SLD files on disk so that it could more easily be used on large programs.

Microsoft

Microsoft's FORTRAN Optimizing Compiler 4.0 was introduced in 1987, replacing an earlier SUBSET compiler, Version 3.3, which had a number of limitations. This compiler supports a myriad of library and command line options in order to be compatible with the earlier products and the full range of memory models of the 8086 CPU. The latest Microsoft product has also added the Codeview FORTRAN source code debugger along with a number of other utility programs.

The compiler is written in C, using the same technology as the Microsoft C compiler. It compiles quite slowly. The code generated for number crunching is faster than average. Version 4.00 was released with a large number (100+) of embarrassing bugs. Several fix disks were issued to correct problems. This latest fixed version is 4.01.04. The compiler performs optimizations, which may be disabled. It creates statics (SAVEd) for all variables and supports large arrays and common blocks. The compiler can generate both listing and cross reference reports. The compiler can also emit an assembly language file. Codeview, the source code debugger, requires a complete recompile and relink in order to use it. Codeview provides a powerful windowing debugger environment, but it is a large program (200K) and keeps tables in memory, significantly limiting its utility on large programs.

The documentation is divided into three separate manuals (*User's Guide*, *Language Reference*, and *Codeview*), each with a separate table of contents and index. The documentation is the most extensive of any of the compilers'. The quality of the documentation is quite good, and indicates in blue ink which features are part of the ANSI 77 standard and are not extensions. The compiler has a useful option that flags all ANSI 77 extensions.

This compiler and its libraries can be the most difficult to install because so many different options are available for memory models, three floating point or emulation libraries, and compatibility with the older compiler. The compiler comes with a modified version of the SETUP utility found with most Microsoft application products. A myriad of features are available for mixed language programming with other Microsoft products.

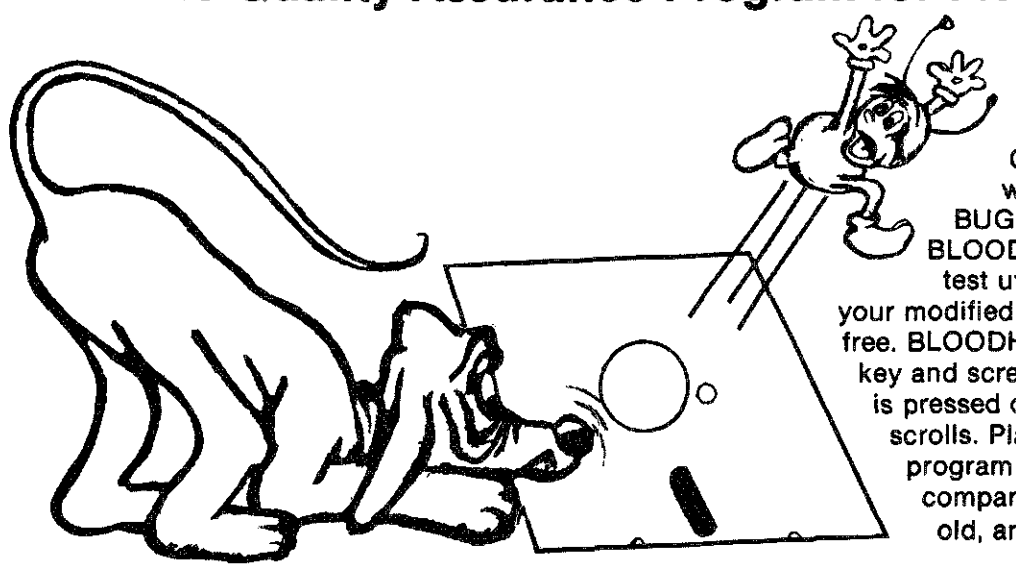
The Microsoft compiler generated the fastest code of any compiler when assumed size arrays were limited to less than 64K, but execution speed dropped substantially when this limitation was removed. Microsoft produced fast times for file I/O. The compiler can also generate code for the 80286/80287 instructions. The code generated was slightly smaller but did not improve execution time. The compiler reports argument type mismatches in subroutine calls over a file. Microsoft has trouble with larger procedures even with optimization turned completely off. I received stack overflow and other errors on large procedures, likely indicating that dynamic table space was exhausted. I also found problems with the I/O library using logical variables to hold characters, a common situation in older FORTRAN code.

Prospero

Prospero Pro FORTRAN-77 was written by Prospero Software of Great Britain and has been sold in the U.S. for some time. Prospero also offers a Pascal compiler under DOS and compilers for the Amiga (68000 CPU) and other machines. Their 68000 products have usually appeared before the 8086 product line. I have assumed that the compiler is written in Pascal and uses the same technology

BLOODHOUND™

The Quality Assurance Program for the IBM PC



Changes in Software often create BUGS. One command to BLOODHOUND, a new test utility, can prove your modified program is error-free. BLOODHOUND captures key and screen whenever a key is pressed or the screen scrolls. Plays back keys after program has been changed, compares new screens to old, and isolates differences.

Price: $129.95. 30 day money back guarantee. Demo disk available.
GOLDBRICK SOFTWARE, 8400 Wells Road, Westminster, CA 92683 (714) 760-9196
"Tools for the lazy programmer"

Number 157 on the Reader Service Card

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

as Prospero's Pascal compiler. The Pro FORTRAN is a two-pass compiler supplied with Prospero's own linker and a source level debugger. It compiles faster than average. The code generated for number crunching is only about average. The compiler creates statics (SAVEd) for all variables and can generate a listing report but no cross reference.

The documentation is divided into a three-part manual ("Overview," "Language Definition," and "Operation"). Each part has a separate table of contents, but only the "Language Definition" has an index. The documentation is less extensive than most of the other compilers'.

The compiler purportedly supports large arrays and common blocks, but the first version failed to execute the Linpack benchmark. The newer version failed to compile it because of expression nesting. Pro FORTRAN was not able to compile large files or files with large procedures. It can report undeclared variables and has options for runtime checks on INTEGER*2 arithmetic and assignment overflow. The compiler can also generate code for the 80286/80287 instructions.

Limited technical support has traditionally been through a distributor in the U.S. I was about to give up on this product. What can you say when you receive the announcement of a new version of the compiler

TASKVIEW

ROCK SOLID MULTITASKING!

Packed with the power you need, TASKVIEW takes you beyond the limits of DOS.

- Communicate while you edit
- Compile while you print
- Load up to 5 megabytes of programs
- Manage resident utilities
- Time-slice multiple jobs
- Cut & paste between programs

TASKVIEW lets you load up to 10 of your favorite applications and switch between them at a keystroke. They can even continue to run while you work on something else! TASKVIEW is the BEST multitasker you can buy. Just ask our customers.

"Thanks for a wonderful product!"

"TASKVIEW is the only multitasker I own which runs efficiently & correctly both on my Zenith Z-151 . . . and on my Orchid 286e . . . I also have DoubleDOS, DESQview, TopView, Windows, and Concurrent PC-DOS'!".

"Fantastic! What DoubleDOS, MS-Windows, and all the other 'stuff' should have been!"

"My BBS is now up 24 hours, 7 days a week thanks to TASKVIEW."

"Thanks! I needed that!"

TASKVIEW requires an IBM PC, XT, AT or Jr compatible, and PC or MS DOS 2.x to 3.x. To get your copy, call toll free:

(800) 367-0651

or send $79.95 + $3.00 S&H ($8.00 Intl.) to:

Sunny Hill Software

PO Box 55278, Seattle, WA 98155-5278

For more information call (206) 367-0650

DoubleDOS trademark Softlogic Solutions, Concurrent PC-DOS reg. trademark Digital Research Corp., MS Windows trademark Microsoft Corp., DESQview trademark Quarterdeck Office Systems, Inc. TopView trademark IBM Corp.

Number 85 on the Reader Service Card

MISOSYS

SOFTWARE FOR YOUR P.C.

Order Now 800-MISOSYS (647-6797)

ED/ASM-86™

- 80X86 Assembly Language development

An integrated editor, assembler, linker, and debugger for 8086, 80186, 80286, and 8087 assembly and disassembly. All features are in a single EXE file.

- a full-featured line EDITOR with intra-line editing, block move and copy, partial save and load.
- a full-screen turbo-type editor is supported on CGA-equipped systems.
- the ASSEMBLER supports structured coding; can directly generate a COM or EXE file; assembly to memory for faster debugging.
- The DEBUGGER includes symbolic disassembly of arbitrary code to disk.

ED/ASM-86 [M-86-030] . . $124.95 + $5 S&H

RATFOR-86™

- Rational FORTRAN translator

A professional implementation of RATional FORtran by James Beard, Ph.D. Provides structure and greater portability to FORTRAN programs to reduce your programming time and effort dramatically over that required when writing directly in FORTRAN. Includes tutorial reference manual.

- compiler has extensive macro support
- "arith" macro for binary arithmetic
- short form READ and PRINT
- enhanced switch case support
- Requires a FORTRAN compiler

RATFOR-86 [M-86-73] . . . $99.95 + $5 S&H

MISOSYS Inc.

P.O. Box 239
Sterling, VA 22170-0239
703-450-4181 MC, VISA, CHOICE
800-MISOSYS/647-6797

VA residents add 4½% sales tax.
S&H: Canada add $1;
Foreign use S&H times 3

Number 121 on the Reader Service Card

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

in June and call the U.S. distributor two months later to find out that they know nothing about a new version. Fortunately, I located the Prospero booth at COMDEX, and they have finally opened a support office in the U.S. I hope the newer versions of this FORTRAN compiler will eliminate the problems I experienced.

WATCOM

The WATFOR-77 compiler is written by WATCOM of Ontario, Canada. This firm specializes in fast FORTRAN programming environments used extensively by educational institutions. Their FORTRAN compilers are available on machines from IBM mainframes to VAX's. The IBM PC version provides an integrated editor, compiler, debugger, and execution environment. The FORTRAN compiler comes in two flavors depending on whether an 80x87 chip is used. WATFOR-77 is an in-memory compiler that compiles fast directly to an EXE file. No object files are created. WATFOR-77 turned in the fastest compile times of the compilers tested. Since no object files are created, the entire source code for a program must be compiled each time. Modular compilation is not supported. Other object modules (libraries for Microsoft FORTRAN) can be dynamically linked during the compilation process. The compiler can generate a listing report but no cross reference. It creates statics (SAVEd) for all variables. WATFOR also comes with an implementation of the Graphics Kernel System (GKS) for CGA, EGA, and Hercules adapters.

The documentation is divided into several perfect bound manuals similar to Borland's (*User's Guide, Language Reference, WATCOM Editor User's Guide, GKS Tutorial and Reference*). Each manual has a separate table of contents and index. The quality of the documentation is good, and the manuals indicate which features are part of the ANSI 77 standard and which are extensions. Like Microsoft's, the compiler can flag all ANSI 77 extensions.

The compiler supports large arrays and common blocks, but it is limited to the available memory space of the in-memory compilation process. The version of the compiler I received did not recognize the 8087 in the presence of the NEC V30 CPU. WATCOM quickly responded with a patch to correct this problem. WATFOR-77 was not able to compile large files or files with large procedures. The compiler can report undeclared variables and has options for built-in runtime checks on integer overflow. For porting purposes, WATFOR-77 would be useful primarily as a quick syntax checker or for flagging ANSI extensions.

Compiler Performance

When we consider performance, we are actually judging a combination of compiler and hardware results. These tests were conducted in two environments: (1) on an AT&T 6300 PC with a Nec V30 CPU and 8087 math chip running at 8 MHz with two 30 megabyte Seagate 238 drives and (2) on a Compaq Deskpro 386-70 with 80387 running at 16 MHz with a fast 70 megabyte hard disk. The AT&T was the primary environment, where all of the tests were executed. Some of the same tests were also executed on the older Compaq 386/287 and the IBM PC for comparison.

In number crunching, the math chip is usually the bottleneck, so execution times for 286 machines are often not that different. The 8087 chip communicates directly on the bus with the 8086 while the 80287 must communicate in a high-speed serial mode with the 80286, which adds some timing overhead. Note that the 80287 chip in most machines runs at 2/3 the speed of the CPU (i.e., a 10 MHz 80286 machine clocks the math chip at 6.7 MHz). Based on the ratio of math chip clock speeds, the AT&T will usually outperform this typical AT machine for number crunching. Execution times on an AT machine with an 8 MHz 80287 are about equal to the 8 MHz 8086/87. For example, the portable Toshiba 3100 (80286 and 80287 chips, both at 8 MHz) clocked the same numeric benchmarks 3% more slowly than the AT&T PC. For an excellent discussion of these math chips, see the articles by Steven Fried in *BYTE*'s annual IBM supplements.

Will the *Real* Benchmark Please Standup?

Several series of FORTRAN programs were used for testing and evaluation. The first tests were based on some standard benchmarking programs: (1) Whetstone and (2) Linpack. Both single precision and double precision varieties are available. The 80x87 chip operates internally on a ten-byte temporary

FORTRAN for the 80386 with source level symbolic debugging

- Use All Available Memory
- Run in the 32-bit Protected Mode of the 80386
- Support for 80287, 80387, and Weitek 1167
- Royalty Free License on Developed Programs

Put the full power of the 80386 into your programs. Our **SVS FORTRAN-386** compiler breaks DOS's 16-bit instruction and 640K memory limit. Cut a typical program's run time in half, while using up to 16 Megabytes of memory. **SVS FORTRAN-386** supports the INTEL 80287 and 80387 and the Weitek 1167 numerical co-processor chips for additional speed. We use IGC's powerful VM/RUN loader and execution manager, included with the compiler, to overcome the DOS limitation.

SVS FORTRAN-386 comes with a powerful **source level symbolic debugger.** The debugger lets the developer view the source codes, access and modify variables by symbolic name, set breakpoints and tracepoints, and reinitialize execution of the program. The debugger requires no modification to the object code, thus letting the user debug the production version of the program. In addition, the program runs under the debugger at full speed until a breakpoint is reached.

Features:

- Full implementation of the ANSI standard FORTRAN77 plus extensions.
- Powerful source level symbolic debugger.
- Comprehensive IEEE floating point software including single precision, double precision and complex arithmetic.
- High-speed compilation generating 32-bit instructions and utilizing all available memory for both code and data.
- Extensive code optimization with inline coding of many intrinsics, optimal register management, loop optimization and more.
- Mature product based on 6 years of development and use.

System Requirements: IBM PS/2 Model 80, COMPAQ Deskpro 386™ and Deskpro 386-20™, Intel Inboard 386™, or other computers based on the Intel motherboard
DOS 2.x or 3.x
INTEL 80287 or 80387, or Weitek 1167 numerical co-processors.

Price:

SVS FORTRAN-386.....$895
SVS C-386.....$895
SVS Pascal-386.....$895
Royalty-free license on developed programs.

Support: Free telephone technical support to registered users. Our support staff has extensive experience in porting and developing programs using SVS FORTRAN-386.

Available from:

SAIC®
Science Applications International Corporation
5150 El Camino Real Suite C-31
Los Altos, California 94022
(415) 960-5931

VM/RUN is a trademark of IGC. COMPAQ and DESKPRO 386 are registered trademark and trademark, respectively, of COMPAQ Computer Corporation. IBM is a registered trademark and Personal System/2 is a trademark of International Business Machines Corporation. Inboard 386 is a trademark of Intel Corporation.

Number **156** on the Reader Service Card

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

real so that the difference between single and double precision is usually small (just loads and stores to the math chip). On mainframe environments, the penalty in time for double precision may be much larger. The second series of programs were actual applications moved from a mainframe environment.

The Whetstone benchmark is a simple minded program that's supposed to test real math and transcendental functions. It is a short synthetic program of about 200 lines (4K). It's been around long enough that reportedly, some unassuming compiler writers have modified their compilers to recognize it and optimize away some of its calculations. Results are reported as the number of Whetstones/second that can be executed. This benchmark should be taken with a large grain of salt. The Linpack benchmark program was written by staff from Argonne Labs. It does matrix inversion using Gaussian elimination on several 200 x 200 matrices and compares the results to the exact results. Both machine precision and speed are tested. This program is more realistic and can't be as easily optimized away. The program size is about 800 lines (22K)—small by mainframe standards. Results are reported as MFLOPS (Millions of FLOating Point instructions per Second) based on the internal calculations. For reference—a Cray executes this benchmark at 12.2 MFLOPS, and both a VAX 11/780 and a micro VAX II under VMS execute at .17 MFLOPS. Double precision versions of this benchmark could not be run on the PC with any of these compilers since the total memory needed exceeded 640K. It could be executed using Lahey F77L-EM on an 80286 or 80386 with extended memory. Those double precision arrays take up lots of space at eight bytes per element.

Table 3 shows the results for these standard benchmarks. All the benchmarks were based on defining integer and logical variables as four bytes. Assumed size arrays were not limited to 64K. You have to be careful to generate consistent benchmarks (i.e., read the code you're compiling). For example, if the Whetstone benchmark is compiled by chance with integers as two bytes (32,767 maximum), several of the loops won't execute the same number of times. Since the loop variables are defined as multiples of 100 times a variable, some large integers (I=100, N4=348*I, N8=899*I, N9=516*I) will just wrap on the 8086 and execute a lot fewer times. The speedy results reported for the Intel and DRI compilers in the *Computer Language* FORTRAN review (January 1986) are such off the wall numbers. Obviously, these compilers don't generate code to trap integer overflow.

Table 3. Standard FORTRAN Benchmarks

	Austec	DRI	Intel	Lahey	Microsoft	Prospero	Watcom	VAX
WHETSTONE (single)								
Compile time	0:57	0:25	0:57	0:17	1:14	0:31	-	
Link time	0:21	1:02	1:31	0:20	0:24	0:38	-	
Total	1:19	1:27	2:28	0:37	1:38	1:09	0:18	
OBJ file size	7,907	10,954	10,789	5,253	8,332	5,722	-	
EXE file size	34,164	57,088	52,895	28,136	30,765	19,200	55,040	
Execution time in k whets/sec								
single prec.	206	151	150	178	160	170	56	1751
double prec.	194	146	144	173	152	165	55	1183
LINPACK								
Compile time	3:11	1:04	Failed	0:38	4:04	Failed	-	
Link time	0:56	1:10		0:36	0:42		-	
Total	4:07	2:14		1:14	4:47		0:40	
OBJ file size	31,553	38,636		23,914	29,472		-	
EXE file size	50,820	137,600		44,578	43,981		390,272	
Execution time in mflops/sec								
single prec.	.030	Died		.017	.023		Crashed	.35

The second test series were smaller FORTRAN programs from a mainframe environment. In this case, the compilers were actually being tested on how well they performed squeezing mainframe code into the PC. These results would of course be affected by the choice of test programs. I tried to choose several programs written by different people that would exercise a variety of FORTRAN features.

Superlite is a ray tracing program developed recently at Lawrence Berkeley Labs (LBL) that calculates the daylighting in a space. Calculations in Superlite are mostly single precision and use transcendentals. Only three lines (INTEGER*4 declarations) were not in conformance with ANSI 77.

Chilc is a nuclear magnetic susceptibility program developed jointly at the University of Oregon and Physico-Technical Institute of Low Temperatures in Kharkov, U.S.S.R. Chilc calculates the magnetic forces in certain kinds of metals using numerical integration with double precision variables. Chilc also complies with the ANSI 77 standard.

Update is a FORTRAN implementation of the CDC UPDATE program for maintaining source code written at LBL. Update uses only integer variables and does lots of file I/O. It is similar in concept to the Source Code Control System available under Unix. Update is representative of much older FORTRAN code, and 10% of its source lines do not conform to the ANSI 77 standard.

These programs are small by mainframe standards (each less than 80K of source). Superlite, however, would run under the PC DOS 640K memory limit only when compiled with most integer and logical variables set to two bytes and one byte, respectively. Mainframe compilers do not need this adjustment, so this comparison favors the DOS compilers for execution times.

The final test was a larger mainframe program. I used the weather processor from

STOP WATCH

MEASURE EXECUTION TIME TO THE MICRO-SECOND

■ ***STOPWATCH™ . . . It Measures Program Time with Microsecond Resolution.*** The new STOPWATCH Program measures total and component times of programs executed on IBM PCs, XTs, ATs, and compatible units to within ± 5 microseconds.

■ ***STOPWATCH . . . It's Easy to Use.*** With STOPWATCH there's no need to modify your program for testing. Simply key ⟨ctl⟩-M instead of ⟨enter⟩ anytime you want to measure execution time to the next display output. (⟨ctl⟩-M and ⟨enter⟩ are equivalent *until* you install STOPWATCH.)

■ ***STOPWATCH . . . Use It To Time New Programs.*** Use STOPWATCH with DEBUG, SYMDEB or CODEVIEW to measure execution time from "GO" to Breakpoint. And, STOPWATCH can be used to time Lotus 1-2-3 Macros, dBASE III Procedures, BASIC Programs, or any object program that you have developed.

■ ***STOPWATCH . . . Use It To Evaluate Your Present System.*** Add STOPWATCH to your present system and measure component times of a program to quickly determine whether a new processor, or faster disk, is the key to improving response time for your application.

■ ***STOPWATCH . . . Unparalleled Accuracy At An Affordable $59.95.*** Included are demo looping programs to demonstrate STOPWATCH's timing ability, the STOPWATCH Program, and a user's manual with easy-to-follow instructions to get you started quickly. Now . . . Microsecond resolution timing — only $59.95! Put your program to the test today with STOPWATCH!

Now Microsecond Resolution Timing Only $59.95 + 3.50 S&H + 6% NJ Tax. MC/VISA accepted

US 800-345-0167
NJ 201-228-7623

CRTS
CUSTOM REAL-TIME SOFTWARE
P O Box 1106, West Caldwell, NJ 07007

IBM is a registered trademark of International Business Machines Corp.
CodeView is a trademark of Microsoft Corp.
Lotus and 1-2-3 are registered trademarks of Lotus Development Corp.
dBase III is a trademark of Ashton-Tate.
*STOPWATCH is a Trademark of Custom Real-Time Software Inc.

Number 124 on the Reader Service Card

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

Table 4. Small Mainframe FORTRAN Program Execution Times

(minutes:seconds)

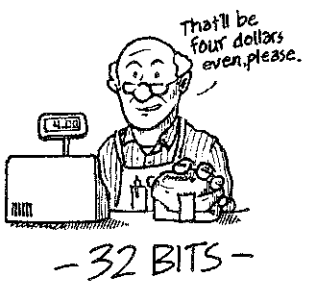
	Austec	DRI	Intel	Lahey	Microsoft	Prospero	WATCOM	VAX
SUPERLITE								
Compile time	Failed	Failed	13:17	4:23	Failed	Failed	Failed	
Link time			2:13	1:00				
Total			15:30	5:23				
OBJ file size			111,986	99,404				
EXE file size			118,508	121,988			161,970	
Execution time			20:47	22:01			235:06	1:53
CHILC (3 files)								
Compile time	5:38	Failed	5:47	1:17	5:51	3:11	-	
Link time	0:37		1:09	0:35	0:41	0:59	-	
Total	6:15		6:56	1:52	6:32	4:10	0:44	
OBJ file size	51,169		59,249	40,393	51,723	48,550		
EXE file size	65,612		83,766	70,144	60,229	56,576	162,848	
Execution time	50:21		73:14	57:51	61:46	138:34	Crashed	8:15
Bad output								
UPDATE (Note 1)								
Compile time	7:14	Failed	5:57	1:30	6:28	2:59	-	
Link time	0:44		1:59	0:42	0:39	0:48	-	
Total	7:58		8:06	2:12	7:07	3:48	1:04	
OBJ file size	69,256		76,444	47,608	55,173	52,400	-	
EXE file size	68,342		92,209	76,040	60,789	51,200	312,656	
Execution time	14:53		6:12	11:39	07:18	Crashed	Crashed	

1 Update was tested with a 160K source deck and a 4K mods file to produce a 450K FORTRAN file and a 250K library file.

DOE2.1C, a large building energy analysis program written at LBL over the last ten years. The weather processor takes a year of weather data, interpolates missing data, produces reports of averages and summaries, and writes a large binary weather file used by the simulators. The weather processor consists of one source file of about 450K (4800 lines). LAHEY F77L was the only DOS compiler able to compile the source file and execute the program properly. All of the other DOS compilers failed to compile the source.

And in the End

You were hoping for a VAX on your desk. Well, not quite, but certainly you're partway there. All of these FORTRAN compilers have specific strengths. As a lowly user, I was not always able to find them. If you are porting from Unix and require recursion, your choices are few—Lahey or Intel. For the serious IBM or VAX porting tasks, the choices are among Austec, Lahey, and Microsoft. I'd also consider Intel despite its failings because of that most wonderful variable and argument error checking across modules at link time. I found Lahey to be the most reliable compiler for the mainframe code I used for testing since it swallows both large procedures and files. Lahey also compiled 4–5 times faster than its competitors, a distinct advantage. Austec and Microsoft have the potential for faster number crunching execution. One thing I can assure you—the more compilers you run your code through, the more you'll learn and the better your code will be.



Austec, Inc.
609 Deep Valley Drive
Rolling Hills Estates, CA 90274
(213) 541-4828

DRI FORTRAN-77
Alexander & Lord
P.O. Box 2110
Carmel Valley, CA 93924
(408) 659-2203

Intel Corporation
Development Tools Catalog
Mail Stop HF2-79
5200 NE Elam Young Parkway
Hillsboro, OR 97214
(800) 874-6835 (U.S. only)

Lahey Computer Systems, Inc.
917 Tahoe Boulevard, Suite 202
P.O. Box 6091
Incline Village, NV 89450
(702) 831-2500

Microsoft Corporation
16011 NE 36th Way
Box 97017
Redmond, WA 98073
(800) 227-4679

Pecan Software Systems, Inc.
1410 39th Street
Brooklyn, NY 11218
(718) 851-3100

Prospero Software, Inc.
100 Commercial Street, Suite 306
Portland, ME 04101
(207) 874-0382

WATCOM Products, Inc.
415 Phillip Street
Waterloo, Ontario
Canada, N2L 3X2
(519) 886-3700

A Hardware Alternative—the Definicon 32-Bit Coprocessing Engines

Definicon Systems is a modest-sized hardware firm that designs and manufactures PC compatible coprocessor boards using the latest 32-bit merchant microprocessor technology. The principals and staff (past and present) include familiar names from the public domain CP/M software world—Trevor Marshall, Dave Rand, and Sigi Kluger. These folks started out as dedicated "hackers" and are now producing some of the most advanced hardware available in the PC environment. Their first product in the summer of 1985 was a coprocessor board using the National Semiconductor 32000 CPU series. In 1986, they came out with boards based on the Motorola 68020 family. Last summer, a new board based on the SUN/3 SPARC RISC chips was announced. The coprocessor board series I reviewed for this article is based on the 68020 CPU and the 68881 FPU chip set, called their DSI-020/DSI-780 series. These are the same chips you'll find in an Apollo or Sun workstation and in the new Macintosh II. Note that Motorola has already started delivering improved versions of these chips, the 68030 and the 68882. The 68882 math chip is pin for pin compatible with the 68881 and should increase math performance by 30 to 70%.

The Hardware

What exactly does Definicon mean by a coprocessor board? The DSI-780 series consists of an IBM PC XT/AT accelerator board that has a self-contained 32-bit computing engine with the following:

- 68020 microprocessor
- 68881 floating point chip
- socket for 68851 memory manager
- from 1 to 16 megabytes of memory
- timers, interrupt and memory controllers
- two serial ports (optional)
- hardware interface to the host PC
- software interface to the PC environment

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

VEDIT PLUS

#1 PROGRAMMABLE EDITOR

Call 1-800-45-VEDIT for FREE Fully Functional Demo Disk

Stunning speed. Unmatched performance. Total flexibility. Simple and intuitive operation. The newest VEDIT PLUS easily satisfies the most demanding computer professional.

Try a Dazzling Demo Yourself.

The free demo disk is fully functional—you can try all features yourself. Best, the demo includes a dazzling menu-driven tutorial—you experiment in one window while another gives instructions.

The powerful "macro" programming language helps you eliminate repetitive editing tasks. The impressive demo/tutorial is written entirely as a "macro"—it shows that no other editor's "macro" language even comes close. And VEDIT PLUS is only 40K in size.

Go ahead. Call for your free demo today, You'll see why VEDIT PLUS has been the #1 choice of programmers, writers and engineers since 1980.

Only VEDIT PLUS is this Flexible.

The installation lets you pick from closely emulating the keyboard layout of Word Perfect, WordStar and others. Or you can easily create your own layout and even your own editing functions. Supports any screen size—you pick screen colors and attributes.

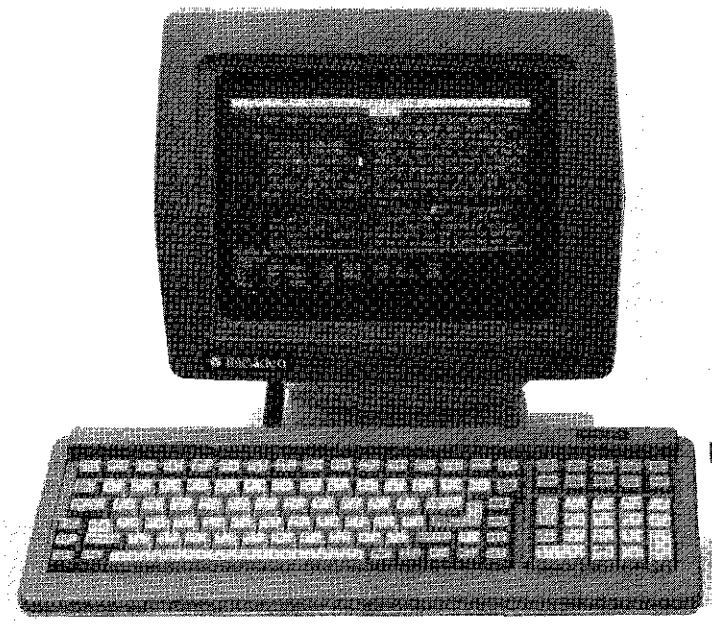
Supports the IBM PC, XT, AT and PS/2. Also supports MultiLink, PC-MOS/386, Concurrent DOS and most networks. Also available for MS-DOS, FlexOS (protected mode), CP/M-86 and CP/M. (Yes, we support windows on most CRT terminals, including CRTs connected to an IBM PC.) Order direct or from your dealer. $185.

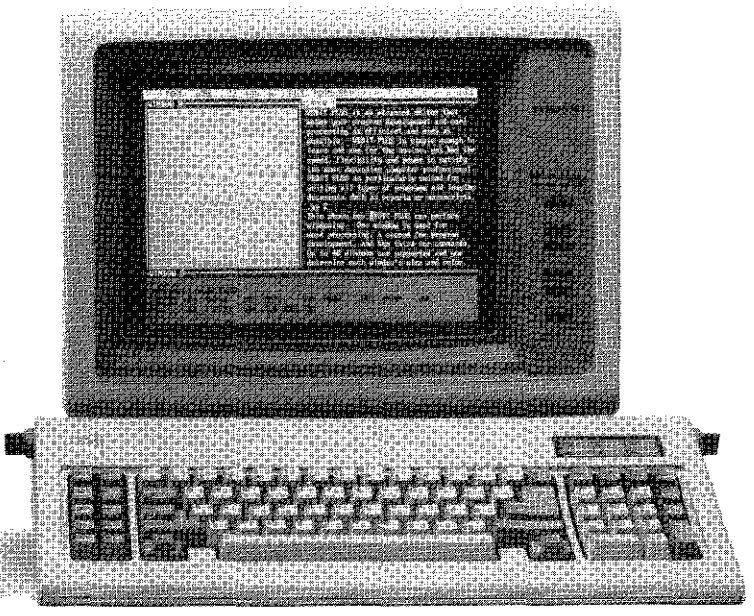
Special: VEDIT (single file, no windows) for CP/M—$49.

- **Fully Network Compatible**
- **Call for XENIX-286 version**
- **30 Day Money-back guarantee**

Compare Features and Speed

	BRIEF	Norton Editor	PMATE	VEDIT PLUS
'Off the cuff' macros	No	No	Yes	Yes
Built-in macros	Yes	No	Yes	Yes
Keystroke macros	Only 1	No	No	Unlimited
Multiple file editing	20+	2	No	20+
Windows	20+	2	No	20+
Macro execution window	No	No	No	Yes
Pop-up menus	No	No	No	Yes
Execute DOS commands	Yes	Yes	Yes	Yes
Automatic processing of Compiler errors	Yes	No	No	Yes
"Cut and paste" buffers	1	1	1	36
Undo line changes	Yes	No	No	Yes
Paragraph justification	No	No	No	Yes
Convert to/from WordStar	No	No	No	Yes
On-line calculator	No	No	No	Yes
Configurable Keyboard	Hard	No	Hard	Easy
43 line EGA support	Yes	No	No	Yes
Manual size/index	250/No	42/no	469/Yes	380/Yes
Benchmarks in 120K File:				
2000 replacements	1:15 min	34 sec	1:07 min	6 sec
Pattern matching search	*20 sec*	*Cannot*	*Cannot*	*2 sec*
Pattern matching replace	2:40 min	Cannot	Cannot	11 sec





VEDIT and CompuView are registered trademarks of CompuView Products, Inc. BRIEF is a trademark of UnderWare, Inc. PMATE is a trademark of Phoenix Technologies Ltd. Norton Editor is a trademark of Peter Norton Computing Inc. MultiLink and PC-MOS/386 are trademarks of the The Software Link, Inc. CP/M and FlexOS are trademarks of Digital Research. MS-DOS is a trademark of Microsoft.

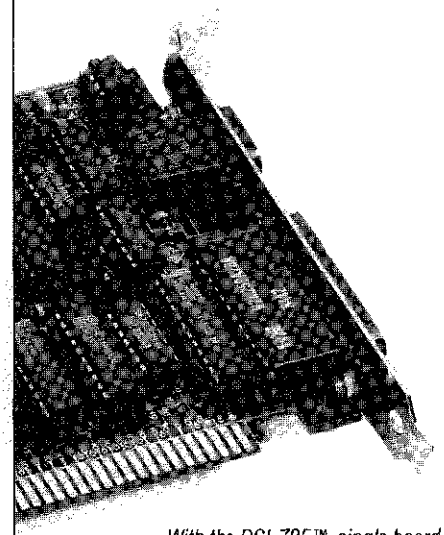
*Also available for TI Professional, Tandy 2000, DEC Rainbow, Wyse WY700 and others.
*Demo disk is fully functional, but does not readily write large files.

CompuView

1955 Pauline Blvd., Ann Arbor, MI 48103
(313) 996-1299, TELEX 701821

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

Mainframe Computing Solutions for your MSDOS™ PC!



With the DSI-785™ single board 32-BIT 68020

The DSI-785™ installs in any IBM™ compatible PC/XT/AT and uses our proprietory MSDOS™ interface software.

Features:

- Motorola 68020 CPU at 16.7, 20 or 25 MHz
- 68881 Floating point coprocessor standard
- Directly address 1, 4, 8, or 16 Megabytes of onboard RAM with only 1 wait state to 25 MHz
- ANSI-77 FORTRAN, PASCAL and C compilers BASIC interpreter
- Multiprocessing MSDOS™ interface
- 30-day money back guarantee

Prices start at under $1500. Call or write now for full details.

Definicon Systems, Inc.
1100 Business Center Circle
Newbury Park, CA 91320 U.S.A.
(805) 499-0652

DEFINICON

MSDOS is a trademark of Microsoft Corporation. IBM is a trademark of International Business Machines Corporation.

Number 153 on the Reader Service Card

Definicon also sells at reasonable prices minicomputer quality optimizing compilers to work with these boards. The pieces missing to complete a working computing engine are peripherals such as floppy and hard disk drives, keyboards, video circuitry, printers, etc. The basic premise for these boards is that they'll take advantage of the low cost of PC based computers and peripherals for these I/O requirements and let the coprocessor do the number crunching.

PC or AT address space. The location of this memory in the PC is adjusted via a jumper setting. Three PC ports are also used to communicate with the 68020 coprocessor. These ports can be used to steer any 64K window of 68020 memory from the 64 megabyte address space to be mapped into the PC memory space. This scheme actually allows support of two DSI boards in an XT machine. The default address space and port options are shown below.

Table 5. Address Space and Port Options

IBM PC Memory Base	IBM PC Ports Used Low Segment	High Segment	Flags
PC XT	E0000 Hex	160 Hex	168 Hex	150 Hex
PC AT	D0000 Hex	2B0 Hex	2B8 Hex	2A0 Hex

The board I received for review was the DSI-020, often called the BYTE Board since it was first introduced jointly with *BYTE* magazine in their July and August 1986, issues. This original board included 1 megabyte of full 32-bit wide dynamic memory (1 wait state) and ran at a clock rate of 12.5 MHz. The board is self-contained with memory refresh logic, interrupt controller and timers, a number of Programmable Array Logic chips (PALs) for interfacing to the PC, and even a patch area. This original board sold for $995. For an extra $100, two serial ports (up to 38,500 baud) and parity check logic were added. An EEPROM that holds a serial number was also added with this upgrade. These boards are typically eight layers with separate ground and power planes, fully socketed. They can be installed in either a standard PC XT or AT slot. If installed in an AT, the boards will take advantage of the 16-bit data bus. The BYTE Board may be upgraded by the factory to a 16.67 MHz clock. Also, Sigi Kluger, one of the Definicon engineers, has written a detailed description of how to add another megabyte of memory to these first boards for the cost of the 36 256K memory chips, some solder, and patience (this description is available on Trevor Marshall's bulletin board, noted later under " Technical Support").

Later versions of this board (the DSI-780 series, named to recall the VAX) are available with up to 25 MHz clock speeds and up to 16 megabytes of 1 wait state memory. For number crunching applications, these faster boards fall near the performance of a VAX 11/750 and 11/780 with floating point accumulators (FPA), so that name appears to be justified. These later versions use Single In-line Memory Modules (SIMMs) soldered to the board for memory, along with 1-megabit chips for the large memory space products. A dynamic memory controller of Definicon's design is used in the newest high speed boards (20 and 25 MHz). With substitution of the newer 68882, these boards should surpass the performance of a VAX.

The hardware interface to the PC is a 64K window of 68020 memory mapped into the

The DSI boards support a maximum of 16 megabytes of user memory. The serial ports, EEPROM, and timers are memory mapped into the address space above 16 megabytes.

A graphics coprocessor board with its own 1 megabyte of memory is also available for the DSI-020/780 using the Intel 82786 32-bit chip. This GRAPHICS-32 coprocessor maps the meg of video RAM into the memory space of the 68020, eliminating the bottleneck of the PC system bus. In conjunction with the 68020 board, it supports bit block transfer rates of 24 million bits/sec and text creation at 25,000 characters/sec. This board supports hardware windowing and panning, 256 colors from a palette of 16 million shades, and built-in graphics primitives on monitors with up to 40 Mbytes/sec video bandwidth.

Definicon Software

All of the DSI boards come with an MS-DOS program loader/interface (20IO) to load 68020 programs onto the board, a minimal assembler, a 68020 linker, assembly language debugger and monitor, utility and diagnostic software, and several public domain programs. The public domain programs include a version of Tiny BASIC for the 68000 and a DSI version of EMACS, the extensible text editor originally written by Richard Stallman. Four additional disks of public domain programs are available for $10.

Software is controlled by the 20IO program loader, a DOS program interfacing to the 68020 software. 20IO supports standard PC hardware devices, configurable DOS device drivers, PC I/O ports, software interrupts, PC memory, and the MS-DOS operating system. The DSI loader uses only 16K of memory at the base of the 68020 memory space for the operating system kernel, PC software interface, and file buffers. All remaining memory is available for user programs. A loader that operates under DRI's Concurrent DOS is available at additional cost for multi-tasking operation. The loader interface is transparent to the user. To start a program on the DSI board, you prefix your

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

Tom Rettig's Library™

Prewritten Solutions to Programming Problems

Clipper Edition

Advanced Extended Library

- Convenient .LIB library file to link
- Separate object files so only the code actually used is added to your program
- Optimized for speed and code size
- Requires Clipper, Winter '85 or later

dBASE III PLUS Edition

Advanced Programmer's Library

- Interface to C/assembler is fully compatible with Clipper's Extend system
- Functions are CALL procedures
- Requires dBASE III PLUS, any version, and 128K of additional memory

Each edition comes complete with...

- Entire source code, nothing withheld. Cleanly written, liberally commented, easy to modify, helpful for learning.
- Support by phone and electronic mail
- No royalties or copy protection
- Documentation inserts for dBASE or Clipper manual. Accurate, complete, easy to use, page-per-command.
- Handy plastic reference card
- Full money-back guarantee

Over 175 functions; 55% written in C, 35% in Assembler, 10% in dBASE

$99.95 per edition at dBASE/Clipper dealers or direct

II/III Tom Rettig Associates
Excellence in dBASE since 1982

9300 Wilshire Boulevard, Suite 470
Beverly Hills, CA 90212-3237, U.S.A.
(213) 272-3784 ▪ Telex: 4996426 RETTIG
Source: BCR480 ▪ CompuServe: 75066,352

dBASE and dBASE III PLUS are Ashton-Tate trademarks, Clipper is a Nantucket trademark

Call or write for free product information

Number **117** on the Reader Service Card

The Heap Expander™ version 2.0

Now your programs can have virtually unlimited heap space using expanded memory, extended memory, disk space, or any combination of the three. And it's all transparent. The Heap Expander's startup code checks the system's resources and uses whatever is available.

still $59.95*

- Uses LIM-standard expanded memory if present.
- Uses AT-style extended memory if present.
- Swaps data to disk as needed.

Libraries and Source Code for:
- Turbo C
- Microsoft C 4.0 and 5.0
- Mark Williams C 3.1 and 4.0
- Lattice C 3.2
- Turbo Pascal 3.0 and 4.0
- Logitech Modula-2/86

Requires an IBM PC, XT, AT, or close compatible with MS-DOS or PC-DOS version 2.0 or above

MC/VISA/COD call
1-800-248-1045 x 100 (US)
1-800-952-5560 x 100 (Idaho)

*Idaho residents add 5% sales tax
Foreign customers add $4.00 for shipping and handling.

The Tool Makers

P.O. Box 8976
Moscow, Idaho 83843
208-883-4979

Number **166** on the Reader Service Card

PORTABLE POWER WITH MasterFORTH

Whether you program on the **Macintosh**, the **IBM PC**, an **Apple II** series, a **CP/M** system, or the **Commodore 64**, your program will run unchanged on all the rest.

If you write for yourself, MasterFORTH will protect your investment. If you write for others, it will expand your marketplace.

Forth is interactive – you have immediate feedback as you program, every step of the way. Forth is fast, too, and you can use its built-in assembler to make it even faster. MasterFORTH's relocatable utilities and headerless code let you pack a lot more program into your memory. The resident debugger lets you decompile, breakpoint and trace your way through most programming problems. A string package, file interface and full screen editor are all standard features. And the optional target compiler lets you optimize your application for virtually any programming environment.

The package exactly matches *Mastering Forth* (Brady, 1984) and meets all provisions of the Forth-83 Standard.

MasterFORTH standard package	$125
(Commodore 64 with graphics)	$100
Extensions	
Floating Point	$60
Graphics (selected systems)	$60
Module relocator (with utility sources)	$60
TAGS (Target Applic. Generation System) – MasterFORTH, target compiler and relocator	$495
Publications & Application Models	
Printed source listings (each)	$35
Forth-83 International Standard	$15
Model Library, Volumes 1-3 (each)	$40

(213) 821-4340

MICROMOTION
8726 S. Sepulveda Bl., #A171
Los Angeles, CA 90045

Number **13** on the Reader Service Card

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

command line with LOAD. In normal operation, you need not distinguish which of the two CPUs (68020 or PC CPU) is actually executing.

What about the real software for this board? Definicon sells Pascal, C, and FORTRAN optimizing compilers and a BASIC interpreter from Silicon Valley Software (SVS). I received the FORTRAN and C compilers for review. These compilers are high quality products from a minicomputer and Unix environment. Definicon also offers a 68020 macro assembler and utilities package from Quelo. Other products include a library manager, a graphics support package, the MKS Unix Toolkit, a Scientific Subroutine library (with FORTRAN source), and a source code symbolic debugger for the SVS language translators. Alternate FORTRAN and Pascal compilers are available with some different strengths and features.

Documentation

The documentation is probably the weakest part of the Definicon products. The DSI boards come with a 110-page *Users Guide and Installation Manual* along with a large 8-1/2" x 11" binder and dividers for other manuals. This documentation is supplemented by various READ.ME and .DOC files on the Loader Disk that comes with each board. Each additional software package such as the FORTRAN and C compilers comes with an additional printed manual and disks with the latest changes or notes. The *Users Guide* has not been updated regularly enough, so the latest on-disk documentation often disagrees with the written manual. The section on installation and hardware is too brief. This is a shame—Definicon staff have written several excellent articles on the DSI hardware in *BYTE*. The *Users Guide* does a better job of explaining the software interface to the DSI loader, but the sections describing the operation of the minimal Assembler, the Linker, and how to use the compilers are much weaker. No index is provided.

The SVS FORTRAN manual consists of about 174 pages with no index. The manual is quite good except that the hardware and operating system specific sections refer to the Unix implementations. These confusing sections are in the last chapter on "Compile Time Options" and in several appendices. The SVS C manual is similar. Definicon needs to supplement these sections with current information on the SVS compilers on the DSI boards.

Technical Support

Technical support for the Definicon hardware and software is provided in several forms. Direct phone support is available during normal business hours along with response to written questions. In addition, a bulletin board is available as a focal point for questions and access to public domain software through Trevor Marshall's Technical Database. A subdirectory on the remote system called 68020\dsi020 has been set up for exchanging public domain software and notes on these DSI boards. Public access via 1200 baud modem is available 24 hours a day at (805) 492-5472 and (805) 493-1495. Definicon also maintains a conference on BIX, the Byte Information eXchange, named DSI.32BIT. I found the on-line technical phone support to be very good.

Installation and Warranties

The board was easy to install since I had no conflicting boards in the machine. I installed the board as it came from Definicon in a slot in an AT&T 6300 PC. This could be more difficult if other boards such as EGA or EMS boards occupy PC memory space above 640K. Remember that the DSI boards need a free 64K of memory window at either D0000 or E0000 (hex). The hardware is warranteed for ninety days parts and labor with a 30-day, money-back return policy.

Performance

Both the C and FORTRAN compilers execute quickly on the DSI boards. I tested these compilers on some standard benchmark programs. Both compilers performed well although I was most impressed with the FORTRAN compiler. SVS FORTRAN compilers are certified by the National Bureau of Standards as conforming with ANSI FORTRAN 77 on hardware by Arete, Raytheon, and Unisys running variants of Unix. The cost of NBS certification would not currently be justified for the DSI operating environment.

SVS FORTRAN conforms with ANSI FORTRAN 77 with a number of mainframe extensions. Some specifics are presented in Table 6. Character constants are limited to 255 bytes. One missing extension is a COMPLEX*16 data type, but this is to be added in the next compiler update. Definicon offers a compiler from Top Express for the DSI boards that supports this data type for such applications.

The SVS compiler supports a variety of compiler options that may be given on the command line when invoking the compiler or embedded in the FORTRAN source code. These options include the following:

- conform to ANSI Standard and flag extensions
- allow Hollerith, character, and numeric data associations
- change default integer size to two bytes, logical to one byte
- change default logical size to two bytes
- limit source code lines to 72 columns
- write binary unformatted files
- pop parameters as in C
- support FORTRAN 66 DO loops
- conditionally compile source lines with D in column 1 (VAX Debug)

The compiler can also generate a cross reference in the listing file and tables of information for the source code debugger and the DSI monitor. SVS supports recursive functions and subroutines, which can be both a blessing and an inconvenience. This feature allows FORTRAN code from many Unix environments that use this feature easy portability to the DSI environment. The one

Fast answers, $189.

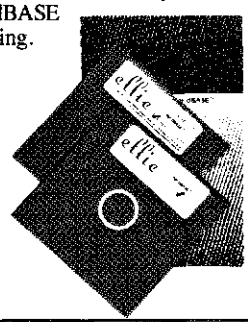
If you use dBASE, we'd like you to know about ELLIE. It's the first English query translator for the relational database standard.

We've *completed* the Information Circuit: IBM provided the hardware... Ashton-Tate the data access software. Until now, that left you with one large task when you needed the answer to a new question. Programming it.

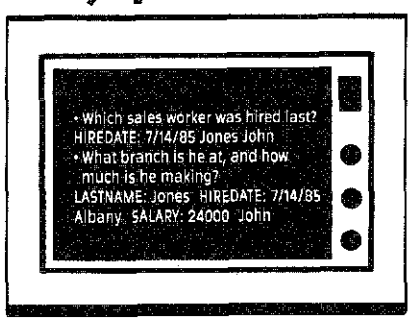
Today you can add the third component of the triad: ELLIE—the English Language Frontend. Now, to get answers from your data, all you do is ask.

Which of the sales workers that completed our advanced training course had higher third-quarter sales than Edwards?

Just ask. ELLIE translates your question into high-quality program (.prg) code, and at the touch of a key flashes control to dBASE for processing.



New! Version 1.1 features advanced memory-residency utilities.


• Which sales worker was hired last?
HIREDATE: 7/14/85 Jones John
• What branch is he at, and how much is he making?
LASTNAME: Jones HIREDATE: 7/14/85
Albany SALARY: 24000 John

dBASE, dBASE III are trademarks of Ashton-Tate Corp.

List the customers that ordered both widgets and wadgets since May. Which American League player with over 300 at bats had the highest on-base percentage? Show the current employer of applicants with BASIC experience, sorting by application date.

After nearly five years of development, how can we offer ELLIE for a mere $189? There are over a million dBASE users. None of them should be without it.

- Auto cross-references up to 5 relational files
- Self-customizing—reads your .dbf files direct
- On-the-fly unit conversion • 512K min. RAM
- For dBASE III/III Plus • (212) 316-9078

ELFSOFT
210 W 101 NY NY 10025

Number 113 on the Reader Service Card

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

missing feature that will likely cause some extra time is the omission of a compiler option to SAVE all local variables and common blocks in static storage. This option is to be included in the next revision. The IBM and VAX FORTRAN compilers don't support recursion, so the FORTRAN SAVE statement is not required. Local variables are used as statics.

Table 6. SVS FORTRAN Extensions

```
Character set:        Upper and lower case are
                      excepted
Source code format:   Column or free format
                      Column length to 120
                      $include supported
Variable names:       Up to 31 characters
Special symbols:      _ and %
Vax-style debug:      D or d in column 1

Extension Data Types Supported
Integer:      INTEGER*1  INTEGER*2  INTEGER*4
Real:         REAL*4     REAL*8
Logical:      LOGICAL*1  LOGICAL*2  LOGICAL*4
Hollerith:    Supported in DATA statements

Recursion:          Fully supported
Intrinsic Routines:
Bitwise         IAND, IOR, INOT, IEOR, and ISHFT
Command-line    GETARG, IARGC
Date and Time   GETTIM, DATTIM, SECNDS, MSEC
Miscellaneous   POKE, IPEEK, RAN
```

The speed of compilation using the 20 MHz board is impressive—zipping along at over 2500 lines per minute on a lowly AT&T PC 6300. This is three times faster than the Lahey F77L FORTRAN Compiler under DOS, one of the fastest DOS compilers. The results of some standard FORTRAN benchmark programs are shown below. For math intensive programs, the 20 MHz DSI board is four to six times faster than a fast PC and nearly twice the speed of a new Compaq 386 with 80387 chip. The 12.5 MHz DSI-020 runs by the fast DOS machines by a factor of three to four. Putting the 80386/387 into the protected mode and using its 32-bit registers changes the picture—however, upgrading the Motorola math chip with the faster 68882 should increase the DSI execution speed by 30 to 70% and temporarily leapfrog past the 80386/387.

As with the DOS compilers, I also compared the results for some actual mainframe programs I recently ported to the DSI board and the IBM PC: Superlite, the ray tracing program developed at Lawrence Berkeley Labs (LBL) that calculates the daylighting in a space, mostly in single precision and using transcendentals; Chilc, the nuclear magnetic susceptibility program developed jointly at the University of Oregon and the Physico-Technical Institute of Low Temperatures in Kharkov, U.S.S.R., which calculates the magnetic forces in certain kinds of metals using numerical integration using double precision variables; and Update, the FORTRAN implementation of the CDC UPDATE program for maintaining source code written at LBL. Update uses only integer variables and does lots of file I/O. It is similar in concept to the Source Code Revision System available under Unix. These programs are small by mainframe standards (less than 80K of source code). Superlite, however, would

FINALLY!

EASYFLOW

An on-screen flowchart processor that knows about flowcharts - not just another "screen draw" program that makes you do most of the work. EASYFLOW is a powerful full-screen graphics program dedicated to flowcharts and organization charts. With it you can quickly compose charts. More important, you can easily modify charts so they are always up to date.

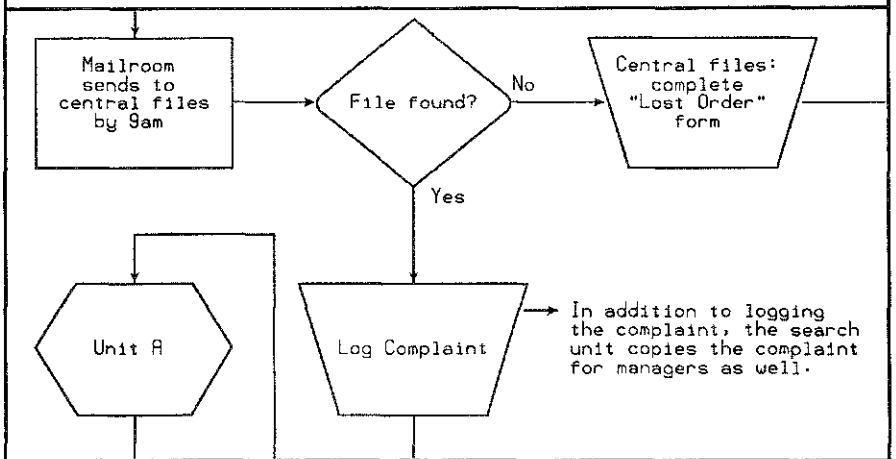
- **Automatic:** Fully automatic text centering within shapes, both horizontally and vertically. Fully automatic line routing & re-routing.
- **Fast:** Written in assembly language for speed.
- **Large:** Charts up to 417 columns wide by 225 lines high. Chart too large for your printer? EASYFLOW automatically breaks the chart up & prints it in page size pieces.
- **Standard:** All standard flowcharting shapes included. Custom shapes can be ordered.
- **User friendly:** Don't take our word for it. PC Magazine* says "EASYFLOW lives up to its name. It's hard to imagine any easier and more flexible way to produce basic and even complex flowcharts".
- **Mouse:** Optional but fully supported.
- **It prints:** On most popular matrix printers including IBM, Epson, Toshiba, HP LaserJet, LaserJet-Plus and many others.
- **It plots:** On HP7440, 7475, 7550, 7570, 7585B and compatible plotters.
- **It works:** We are contractually prevented from mentioning the name of the "big eight" accounting firm that purchased a world-wide site license, but we can tell you that they spent months evaluating all available flowcharting packages before choosing EASYFLOW.
- **Rush delivery:** Order by noon today (eastern time) and we'll have it to you by courier tomorrow**. Rush delivery charge is $15.00 (instead of $2.00) and is available only in USA & Canada.
- **Documented:** 100 page manual plus over 150 screens of context sensitive help.

EASYFLOW works on IBM PC's, IBM PS/2 and compatibles. Requires 384 K memory, DOS 2.0 or higher and an IBM CGA/EGA/VGA or Hercules monochrome compatible adapter card.

Order direct for only **$149.95** + $2.00 S&H (USA/Canada), $10.00 (foreign). Payment by M.O., cheque, VISA, Mastercard or Company PO.

* March 10, 1987 issue, page 278.

** Rush orders are shipped by Purolator Courier and normally arrive the next business day to most locations. Remote destinations take longer.


Mailroom sends to central files by 9am
File found?
No
Central files: complete "Lost Order" form
Yes
Unit A
Log Complaint
In addition to logging the complaint, the search unit copies the complaint for managers as well.

The chart fragment above was produced on an HP LaserJet-Plus and is actual size and unretouched. Publication quality charts like this can be produced using only minutes of preparation time and seconds of print time.

HavenTree Software Limited
PO Box 1093-V
Thousand Island Park, NY 13692

Order Desk: 1-800-267-0668
Information: (613) 544-6035 ext 31
Telefax(G3): (613) 544-9632

Number **154** on the Reader Service Card

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

MetaWINDOW
Power Graphics for your PC!
PC TECH JOURNAL
"Product of the Month"
"... a technological tour de force for fast PC graphics."
NO ROYALTIES!
NEW! - QuickWINDOW/C
All the features of MetaWINDOW for Microsoft Quick/C! - $95*
Unparalleled Performance!
MetaWINDOW provides an expanded set of graphic drawing functions, plus the added functionality and performance required for designing multi-window desktop applications.
• auto-cursor tracking
• pull-down menus
• pop-up windows
• comprehensive graphic functions
• multiple fonts
Enhanced Features!
• Display multiple bitmap or "filled-outline" fonts.
• Face fonts for bold, italic, underline or strike-out stylings.
• Full "RasterOp" transfer functions for writing, erasing, rubberbanding or dragging: lines, text, icons, bit images and complex objects.
• Create pop-up menus, windows and icons.
• Supports IBM's new PS/2 VGA and MCGA graphics.
MetaWINDOW comes complete with language bindings for 20 popular C, Pascal and Fortran compilers, plus dynamic runtime support for over 50 graphics adaptors and input devices.
MetaWINDOW
Advanced Graphics Toolkit
4 disks, 3 260 page manuals - $195*
NEW! - TurboWINDOW/Pascal
All the features of MetaWINDOW for Borland Turbo Pascal Ver. 4! - $ 95*
* Plus $5.00 shipping and handling
TO ORDER CALL 1-800-332-1550
For information or in CA call 408-438-1550
METAGRAPHICS
SOFTWARE CORPORATION
269 Mount Hermon Road
Scotts Valley, CA 95066
Number 143 on the Reader Service Card

Table 7. Standard FORTRAN Benchmarks

Hardware/Software				Whetstone (1000/sec) single	Whetstone (1000/sec) double	Linpack (MFLOPS) single	Linpack (MFLOPS) double	
DSI-780+	20.0 MHz	SVS		1285	1135	.13	.11	
DSI-020	12.5 MHz	SVS		792	708	.08	.07	
VAX 11/785 w/FPA		VMS 4.1		1751	1183	.35	.14	
Compaq 386/387	16.0 Mhz	386 DOS		1422		.19	.15	See Note 1
Compaq 386/387	16.0 MHz	DOS 3.2		765	700	.06	--	
Compaq 386/287	16/8 MHz	DOS 3.1		306	283	.036	--	
AT&T 6300/8087	8.0 MHz	DOS 3.1		206	194	.031	--	
IBM-PC/XT/8087	4.77 MHz	DOS 2.1		94	86	.015	--	

1 The fast 80386 times for Whetstones were based on a Beta test copy of NDP386 FORTRAN by Microway (Green Hills compiler) running under Phar Lap 386 DOS Extender in the protected mode. The fast 80386 times for Linpack were based on a Beta test copy of SVS FORTRAN (sold by SAIC) running under the IGC X-AM DOS extender.

2 The PC times are based on Austec RM FORTRAN 2.42 for Whetstone and Linpack. These were the best times of the DOS compilers tested.

run under the PC DOS 640K memory limit only when compiled with most integer and logical variables set to two bytes and one byte, respectively. The mainframe and SVS FORTRAN compilers do not need this adjustment, so this comparison favors the DOS compilers for execution times.

My normal work consists of using hour-by-hour building simulation analysis programs that model the energy use in large buildings. I am in the process of porting two large research programs (DOE2.1C and BLAST3.0) to a Prime minicomputer and debugging a new version of DOE2 on an IBM 3083 mainframe. DOE2.1C was developed at Lawrence Berkeley Labs over the last ten years with assistance from other federal labs and outside contractors. DOE2 source code tape and test files are about 35 megabytes. BLAST, developed for the Army Corps of Engineers over about the same time span, is now maintained by staff at the University of Illinois. Both of these programs have embodied several hundred years of software development not easily duplicated. Ironically, both of these programs were initially written for a CDC machine. Both programs are currently supported on both CDC and VAX computers, and DOE2 on IBM mainframes.

I used the weather processors from these larger programs as a test for porting larger applications. The source code used for testing were the VAX versions. The weather processors take a year of hourly weather data at a site, interpolate missing data, produce reports of averages and summaries, and write a large binary weather file used by the simulators. The DOE2 Weather processor consists of one file of about 450K of source code (4,800 lines). The SVS compiler flagged several errors with ENCODE statements that were easily remedied, and the code compiled quickly onto the DSI board; however, it did not execute properly this first cycle. Since SVS supports recursion, common blocks first defined in a subroutine had to be SAVEd in static storage. The foolproof way to handle this is to add a SAVE statement to each subroutine and function—a lot of editing. After looking over the source code, I moved all the common block definitions into the MAIN routine, which had the same effect. I accomplished this by using a find (grep) utility to write all common blocks to

Table 8. Small Mainframe FORTRAN Program Execution Times

Hardware/Software				Superlite (min:sec)	Chilc (min:sec)	Update (1) (min:sec)	
DSI-780+	20.0 MHz	SVS		4:15	9:39	5:45	
DSI-020	12.5 MHz	SVS		6:53	15:01	6:53	
VAX 11/785+ w/FPA		VMS 4.1		1:53	8:15		
Compaq 386/387	16.0 MHz	386 DOS		4:51	10:33		See Note 2
Compaq 386/387	16.0 MHz	DOS		6:16	15:08		
Compaq 386/287	16/8 MHz	DOS		18:28	36:38		
AT&T 6300/8087	8.0 MHz	DOS		20:47	50:21	6:12	
IBM PC/XT/8087	4.77 MHz	DOS		45:07	111:46		

1 Update was tested with a 160K source deck and a 4K mods file to produce a 450K FORTRAN file and a 250K library file.

2 The fast Compaq 386/387 times were with SAIC's SVS FORTRAN for Superlite and with MicroWay's NDP386 FORTRAN running in the 386 protected mode.

3 The PC times are based on Intel FORTRAN-86 for Superlite and Update, and Austec RM FORTRAN 2.42 for Chilc. These were the best times of the DOS compilers tested.

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

a file, removed duplicate entries, and inserted this in MAIN. The total edit, compile, link, and test cycle until operating was fewer than three hours. Only one DOS FORTRAN compiler (Lahey) out of eight could compile and execute this same DOE2 weather processor properly. The BLAST weather processor WIFE consists of one small main file, several large files of subroutines, and a score of small include files. The SVS Compiler handled this job well, too.

I also compiled the VAX version of the DOE2 syntax checker (about 35,000 lines in one source code file of over 2 megabytes) using the SVS FORTRAN compiler. This required compiling and executing several utility programs in order to create data files and libraries for the syntax checker. I experienced very little difficulty with this task. The SVS FORTRAN compiler gave better diagnostics and errors than either the VAX or IBM mainframe compilers. The SVS compiler caught as errors those cases where arguments were mismatched from called to calling subroutines. In over 35,000 lines of code originally from the strange CDC environment, the SVS compiler posted only a couple pages of errors that weren't warning of incorrect argument passing. These errors were VAX TYPE statements, ENCODE and DECODE statements, and using Boolean operations (AND and OR) on integer variables. Using the ANSI FORTRAN 77 flag on either the VAX or SVS compiler indicated that over 10% of the lines of code did not conform to the ANSI 77 Standard.

The Bottom Line

I was impressed enough from my review of the original DSI board and the available compilers to buy one myself. Using the 30-day, money-back guarantee period, I recently traded up for a DSI-780+ board with 2 megabytes. I'm waiting for the new 68882 math chip to increase execution speed. One of these days, I might just write a Macintosh emulator that runs on a PC. Until then, this board should at least provide me the ability to port large mainframe software to the PC without having to bust one of those proverbial #@? 64K segments. ♦

Definicon Systems, Inc.
1100 Business Center Circle
Newbury Park, CA 91320
(805) 499-0652

The test files used for this article are available on PJ's bulletin board services and the Issue Disk for Volume 6.1.

M. Steven Baker holds degrees in architecture and electrical engineering from the Massachusetts Institute of Technology and M. A. degrees in architecture and urban and regional planning from the University of Oregon. Steve works as an energy consultant for the Oregon Department of Energy.

Escape

...through the windows of *Show Me!*

F1-OPEN F2-HELP F3-DIR F4-SEARCH F5-POSITION F6-SWITCH Esc-EXIT V3.00

Escape that caged-in feeling you often experience as you code and debug programs — view files anytime with *Show Me!*, the memory-resident file viewing power tool. There's nothing like it! Try *Show Me!* and you'll never write another line of code without it!

Technical Highlights

- view up to four files at once
- create layered or tiled windows
- paste / print all or part of a file
- display files in hexadecimal
- search for strings or hex values
- view files of unlimited size
- scroll vertically & horizontally
- scroll all windows simultaneously
- list directories inside a window
- read WordStar document files
- memory-resident or use from DOS
- requires as little as 36K RAM

"***Show Me! is great!***"
"It has greatly improved my program development efficiency... Thanks for the great work!"
— Skip Douglas, Lisle, IL

" ... discover the wealth *Show Me!* puts at your fingertips."
— InfoWorld, 11/23/87

SHOW ME! «ITSGREAT» Line 1 Col 1

Show Me!

New Version III
*Only $39**

No Risk 30-Day Trial

For Information or To Order Call (800) 634-3122

*Introductory Price
($59.95 after 3/1/88)
Add $5 shipping/handling;
TX orders add 6% sales tax.

Serengeti Software
P.O. Box 27254 • Austin, TX 78755
(512) 345-2211
Show Me! is a trademark of Serengeti Software

Ask about *So Help Me!*, the windowing help screen driver to add full-color, disk resident help windows to the programs you write — royalty free — for only $79.

Number 161 on the Reader Service Card

Definicon Pricing: 68020 Hardware

DSI-020	1 Megabyte RAM, with CPU & FPC at 12.5 MHz	$1,094
DSI-780+/2	2 Megabyte RAM, with CPU & FPC at 20.0 Mhz	$2,695
Note that many other configurations are available		
GRAPHICS-32	Graphics Adaptor for DSI Boards	$1,295
68020 Software		
SV20-F	Silicon Valley Software ANSI FORTRAN 77	$ 528
SV20-C	Silicon Valley Software "C" Compiler	398
SYM-20	Symbolic Debugger	299
SSP20-S	Scientific Tool Box with Fortran Source	314
QL20-MA	Quelo Macro Assembler and Utilities	198
PD20	Public Domain Diskettes (4)	20

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

C CODE FOR THE PC

source code, of course

C Source Code

Bluestreak Plus Communications (two ports, programmer's interface, terminal emulation)	$400
Greenleaf Data Windows (windows, menus, data entry, interactive form design)	$315
Barcode Generator (specify Code 39 (alphanumeric), Interleaved 2 of 5 (numeric), or UPC)	$300
GraphiC 4.0 (high-resolution, DISSPLA-style scientific plots in color & hardcopy)	$275
Vitamin C (MacWindows)	$200
Greenleaf Communications Library (interrupt mode, modem control, XON-XOFF)	$160
Greenleaf Functions (296 useful C functions, all DOS services)	$160
Essential C Utility Library (400 useful C functions)	$160
Essential Communications Library (C functions for RS-232-based communication systems)	$160
PC/IP (CMU/MIT TCP/IP implementation for PCs)	$100
B-Tree Library & ISAM Driver (file system utilities by Softfocus)	$100
The Profiler (program execution profile tool)	$100
Entelekon C Function Library (screen, graphics, keyboard, string, printer, etc.)	$100
Entelekon Power Windows (menus, overlays, messages, alarms, file handling, etc.)	$100
QC88 C compiler (ASM output, small model, no longs, floats or bit fields, 80+ function library)	$90
CBTree (B+tree ISAM driver, multiple variable-length keys)	$80
ME (programmer's editor with C-like macro language by Magma Software)	$75
Wendin PCNX Operating System Shell	$75
Wendin PCVMS Operating System Shell	$75
Wendin Operating System Construction Kit	$75
EZ_ASM (assembly language macros bridging C and MASM)	$60
Multi-User BBS (chat, mail, menus, sysop displays; uses Galacticomm modem card)	$50
Heap Expander (dynamic memory manager for expanded memory)	$50
Make (macros, all languages, built-in rules)	$50
Vector-to-Raster Conversion (stroke letters & Tektronix 4010 codes to bitmaps)	$50
Coder's Prolog (inference engine for use with C programs)	$45
PC/MPX (light-weight process manager; includes preemption and cooroutine packages)	$45
Biggerstaff's System Tools (multi-tasking window manager kit)	$40
TELE Kernel (Ken Berry's multi-tasking kernel)	$30
TELE Windows (Ken Berry's window package)	$30
Clisp (Lisp interpreter with extensive internals documentation)	$30
Translate Rules to C (YACC-like function generator for rule-based systems)	$30
6-Pack of Editors (six public domain editors for use, study & hacking)	$30
ICON (string and list processing language, Version 6 and update)	$25
LEX (lexical analyzer generator)	$25
Bison & PREP (YACC workalike parser generator & attribute grammar preprocessor)	$25
C Compiler Torture Test (checks a C compiler against K & R)	$20
PKG (task-to-task protocol package)	$20
A68 (68000 cross-assembler)	$20
Small-C (C subset compiler for 8080 and 8088)	$20
tiny-c (C subsubset interpreter including the tiny-c shell)	$20
Xlisp 1.5a (Lisp interpreter including tiny-Prolog in Lisp)	$20
List-Pac (C functions for lists, stacks, and queues)	$20
XLT Macro Processor (general purpose text translator)	$20
C Tools (exception macros, wc, pp, roff, grep, printf, hash, declare, banner, Pascal-to-C)	$15

Data

DNA Sequences (GenBank 48.0 of 10,913 sequences with fast similarity search program)	$150
Protein Sequences (5,415 sequences, 1,302,966 residuals, with similarity search program)	$60
Webster's Second Dictionary (234,932 words)	$60
U. S. Cities (names & longitude/latitude of 32,000 U.S. cities and 6,000 state boundary points)	$35
The World Digitized (100,000 longitude/latitude of world country boundaries)	$30
KST Fonts (13,200 characters in 139 mixed fonts: specify TEX or bitmap format)	$30
NBS Hershey Fonts (1,377 stroke characters in 14 fonts)	$15
U. S. Map (15,701 points of state boundaries)	$15

The Austin Code Works
11100 Leafwood Lane
Austin, Texas USA 78750-3409

Voice: (512) 258-0785
BBS: (512) 258-8831
Email: FidoNet 1:382/12

Free surface shipping on prepaid orders

MasterCard/VISA

Number 30 on the Reader Service Card

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

A Five-Year Index of *Programmer's Journal* Articles

This cumulative index was prepared in large part by employees of Macon Software using their product, Advanced DB Master. We are grateful for the effort put forth and can recommend Advanced DB Master for simple database needs. We found it easy to input and update the files, and the report generator is simple, powerful, and easy to use. Advanced DB Master is available from Macon Software, 724 S. Tejon, Colorado Springs, CO 80903.

Department	Date	Vol.	Page	Author	Title
ASM	4/83	1.2	4	Whitman, David	Writing an Assembler for the IBM PC
ASM	6/83	1.3	16	Zachai, Art	Programming in Assembly Language
ASM	6/83	1.3	32	Rollins, Dan	ASM Line Drawing Program
ASM	8/83	1.4	11	Zachai, Art	Beep & Beepint Stay Resident
ASM	8/83	1.4	27		BIOS Equates
ASM	10/83	1.5	20	Zachai, Art	5 Ways to Combine Modules
ASM	8/83	1.4	18	Rollins, Dan	While/Wend
ASM	12/83	1.6	17	Millhollin, Rick	Printer Configuration in Assembly
ASM	12/83	1.6	27	Rollins, Dan	Flying Saucers in Assembly
ASM	12/83	1.6	36	Kaplon, Morton F.	Disk Parameter Equates
ASM	2/84	2.1	32	Reuss, Ted	Assembly Modules for C
ASM	4/84	2.2	20	Auerbach, Joshua	Assembly Language Tutorial
ASM	6/84	2.3	17	Auerbach, Joshua	Assembly Language Tutorial, Part 2
ASM	8/84	2.4	21	Burk, Ron	8088 Assembly Macros
ASM	10/84	2.5	29	Rayhawk and Radzio	Screen I/O Routines
ASM	12/84	2.6	23	Salkin, Bill	Memory Segmentation
ASM	2/85	3.1	14	Salkin, Bill	Memory Segmentation Examples
ASM	4/85	3.2	22	Rollins, Dan	The Ask/Batch File Utility
ASM	6/85	3.3	19	Hoyt, Michael L.	Printer Control from a Batch File
ASM	8/85	3.4	23	Jackson, John W.	How to Cure the "Dreaded Locked Keyboard" Condition
ASM	10/85	3.5	20	Hoyt, Michael	A Menu-driven PC Using display.com
ASM	4/86	4.2	32	Hoyt, Michael	Optimizing for Speed
ASM	6/86	4.3	32	Defenbaugh, George	Memory Model Independence in Assembler
ASM	6/86	4.3	38	Hawkinson, C. D.	An Approach to Windowing for the Assembly Language Programmer
ASM	8/86	4.4	36	Abrash, Michael	More Optimizing for Speed
ASM	10/86	4.5	40	Daetwyler, Dan	Search and Display
ASM	10/86	4.5	44	Pearson, Graham	An Assembler Programming Suggestion
ASM	4/87	5.2	26	Baker, M. Steven	A Graphics Print Screen Utility: A Simple Terminate and Stay Resident Program
ASM	4/87	5.2	34	Pearson, Graham	Coding Guidelines for Resident Programs
ASM	10/87	5.5	24	Swan, Tom	The Case of the Invisible CapsLock
BASIC	2/83	1.1	22	Jones, D.	Rounding
BASIC	4/83	1.2	23	Minasi, Mark	Intro to User Defined Character Sets
BASIC	8/83	1.4	20	Rayhawk, Joe	Some Suggestions for Coding in IBM PC/BASIC
BASIC	10/83	1.5	23	Rayhawk, Joe	Some Suggestions for Coding in IBM PC/BASIC, Part 2
BASIC	12/83	1.6	37	Estes, Greg	Array and File Handling
BASIC	2/84	2.1	29	Moak, Ronald	Compiled BASIC--the Next Step
BASIC	4/84	2.2	16	Buchanon, Wat	Optimizing BASIC Compiler Source Code
BASIC	4/84	2.2	31	Moak, Ron	Full Screen Input Routines
BASIC	6/84	2.3	24	Moak, Ronald	Graphic Windows
BASIC	8/84	2.4	25	Moak, Ronald	Random Access Files in BASIC, Part 1
BASIC	10/84	2.5	17	Thompson, Karl F.	Scroll Control Using BASIC and Assembler: Adding Finesse to Your BASIC Programs
BASIC	10/84	2.5	36	Moak, Ronald	Header Records in Random Access Files, Part 2
BASIC	12/84	2.6	17	Gazdar and Shroff	Calendar Functions for BASIC
BASIC	2/85	3.1	20	Moak, Ron	Index Files, Part 1
BASIC	2/85	3.1	24	Laswell, Bette	A Commercial Programmer's Look at BASIC on the IBM PC
BASIC	4/85	3.2	26	Moak, Ronald	Index Files, Part 2
BASIC	4/85	3.2	33	Robinson, Bernard H., Jr.	Calendar Date Routines in BASIC
BASIC	6/85	3.3	33	Duke, Melvin O.	Repositioning the Stars in Starfinder
BASIC	12/85	3.6	41	Jackson, John W.	A Better Way to Interface BASIC and Assembly
BASIC	4/86	4.2	43	Robinson, Bernard H., Jr.	Structuring BASIC Output with Field Statements
BASIC	4/86	4.2	48	Shaffer, Robert R.	Networking--What's the Problem?
BASIC	8/86	4.4	50	Robinson, Bernard H., Jr.	Tips on Formatting with BASIC PRINT USING Statements
BASIC	2/87	5.1	56	Robinson, Bernard H., Jr.	More Tips on Formatting BASIC Data
BASIC	4/87	5.2	54	Winer, Ethan	Indexing in BASIC
BASIC	6/87	5.3	54	Dymerski, Paul P.	Formatting BASIC Output with Buffered PRINT USING
BASIC	8/87	5.4	50	Robinson, Bernard H., Jr.	Variable Passing and Hiding in QuickBASIC 2 Subprograms
BASIC	10/87	5.5	52	Dymerski, Paul P.	A Matchup of Turbo Basic 1.0 and QuickBASIC 3.0
C	4/83	1.2	27	Randall, James E.	Accessing the BIOS Functions from C86
C	8/83	1.4	14	Hailey, Bill	Joystick Interface Routines, C86
C	10/83	1.5	18	Smolek, Ken	C Transportability Problems
C	12/83	1.6	24	Keating, Ed	C the Difference
C	2/84	2.1	13	Smolek, Ken	Sorting in C
C	4/84	2.2	10	Killan, Don	C What's on the Stack
C	4/84	2.2	27	Reuss, Ted A.	Date, Time and Free Space
C	4/84	2.2	35	McCrory, Robert T.	Music in C
C	6/84	2.3	12	Cabito, Steve	Databurst: Mainframe-style Screen Management
C	6/84	2.3	28	Wiley, Christopher	PC Musician, Creating a Music Composition and Editing System
C	8/84	2.4	29	Hillman, Warren	C Utility Routines
C	10/84	2.5	13	Rollins, Dan	The C Bandwagon

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

Department	Date	Vol.	Page	Author	Title
C	12/84	2.6	15	Jaeschke, Rex	Field Usage and Pointer Subtraction in C
C	2/85	3.1	12	Greco, Frank D.	Using the C Preprocessor
C	4/85	3.2	20	Jaeschke, Rex	Doctor C's Tips--Pointers to Functions
C	6/85	3.3	13	Volkstorf, J. Edward, Jr.	Instant-C
C	8/85	3.4	21	Moak, Ron	C Screen Functions, Part 1
C	10/85	3.5	17	Moak, Ron	C Code Excerpts, Part 2
C	12/85	3.6	28	Swan, Randall B.	Common C Bugs
C	2/86	4.1	10	Swan, Randall B.	Autoinc Explained: Evaluating the Auto-increment with Respect to Addressing and Dereferencing
C	2/86	4.1	14	Ackerman, Robert	Sorting Using an AVL Tree
C	4/86	4.2	30	Jaeschke, Rex	The Proposed ANSI C Language Standard
C	6/86	4.3	42	Jaeschke, Rex	The Proposed ANSI C Language Standard, Part 2
C	6/86	4.3	47	Hunt, William J.	Microsoft 3.00 and Lattice 3.00
C	8/86	4.4	44	Hunt, William J.	Memory Resident Programs in C
C	10/86	4.5	46	Jaeschke, Rex	The Proposed ANSI C Language Standard, Part 3
C	10/86	4.5	48	Hunt, William J.	Monitoring Stack Usage and a Guide to Startup Source Code
C	12/86	4.6	40	Hunt, William J.	Alignment Problems in C and Managing Memory Allocation
C	12/86	4.6	44	Jaeschke, Rex	The Proposed ANSI C Language Standard, Part 4
C	2/87	5.1	52	Jaeschke, Rex	The Proposed ANSI C Language Standard, Part 5
C	4/87	5.2	48	Hunt, William J.	Portability in Practice and Using Header Files
C	4/87	5.2	50	Jaeschke, Rex	The Proposed ANCI C Language Standard, Part 6
C	6/87	5.3	46	Jaeschke, Rex	The Proposed ANSI C Language Standard, Part 7
C	6/87	5.3	50	Hunt, William J.	Coping with ANCI C and Forcing File Updates
C	8/87	5.4	38	Jaeschke, Rex	The Proposed ANSI C Language Standard
C	8/87	5.4	42	Hunt, William J.	Character Tables and String Functions
C	10/87	5.5	44	Jaeschke, Rex	The Proposed ANSI C Language Standard: limits.h and float.h Headers
C	10/87	5.5	48	Robinson, Bernard H., Jr.	A Note on Printer Output from Turbo C Programs
C	12/87	5.6	60	Hunt, William J.	Handling Fixed Length Character Arrays
C	12/87	5.6	62	Jaeschke, Rex	The Proposed ANSI C Language Standard
COBOL	2/83	1.1	11	Rohde, Rich	An Evaluation of IBM's COBOL
Consulting	10/85	3.5	7	Barkley, Paul	Consultants and Other Folk
Consulting	12/85	3.6	14	Barkley, Paul	How to Recover All Your Costs
Consulting	4/86	4.2	12	Barkley, Paul	The Home Office Syndrome
Consulting	6/86	4.3	21	Barkley, Paul	Reusable Modules
Consulting	8/86	4.4	18	Barkley, Paul	Efficient Custom Programming
Consulting	10/86	4.5	37	Barkley, Paul	Tax Traps, Consulting Styles, and Nest Eggs
Consulting	2/87	5.1	20	Barkley, Paul	Marketing and Rules to Live By
Consulting	6/87	5.3	24	Barkley, Paul	Merger Mania
Consulting	8/87	5.4	22	Barkley, Paul	Consultant Productivity and Effectiveness
Consulting	10/87	5.5	20	Barkley, Paul	Making Computer Tools Pay for Themselves
Consulting	12/87	5.6	20	Barkley, Paul	How to Get Rich in Consulting
Copy Protect	4/83	1.2	21	Anderson, Timothy S.	Copy Protection
Copy Protect	8/84	2.4	17	Larratt, Richard	The Future of Disk Protection
DBase II	2/83	1.1	18	Detesto, Frankie	First Impressions
DBase II	4/83	1.2	30	Detesto, Frankie	DBase II
DBase II	6/83	1.3	35	Detesto, Frankie	DBase II
DBase II	8/83	1.4	26	Detesto, Frankie	Report Time
DBase II	10/83	1.5	30	Ridder, Daniel	Bargraph Programs for DBase II
DBase II	12/83	1.6	39	Bohle, Jim	Nike Marathon Race Results with DBase II
DBase II	2/84	2.1	34	Bohle, Jim	Database Relations--Select, Locate, Index and Find
DBase II	4/84	2.2	40	Bohle, Jim	A Printer Control Database
DBase II	6/84	2.3	35	Bohle, Jim	Creating Screens
DBase II	8/84	2.4	38	Gowens, John W.	DBase II/Knowledgeman/R:Base--What Database Manager Should You Really Buy?
DBase II	10/84	2.5	41	Hillman, Warren	Calling Assembly Routines from DBase II
DBase II	12/84	2.6	39	Moak, Ronald	DBase II to DBase III
DBase II	10/85	3.5	34	Salter, Brett	DBASE II Routines
DBase II	12/85	3.6	38	Salter, Brett	More DBASE II Routines
DBase II	2/86	4.1	42	Salter, Brett	The Final Installment of Brett Salter's dTools
DBase II	8/86	4.4	57	Liskin, Miriam	The Structure of DBASE II and III
DOS	8/83	1.4	8	Kaplon, Morton F.	Impressions of DOS 2.0
DOS	10/83	1.5	13	Detesto, Frankie	Multitasking (products)
DOS	12/83	1.6	20	Salter, Brett	The DOS Patch Utility
DOS	8/86	4.4	22	Pearson, Graham	DOS 2 File Handles and File I/O
DOS	12/86	4.6	22	Defenbaugh, George	Parents, Children, Redirection, and Piping with DOS Functions 45H and 46H
DOS	12/86	4.6	26	Baker, Steven	Terminate and Stay Resident
DOS	2/87	5.1	24	Greco, Frank D.	Redirection, Or "They Went That-a-way"
Hardware	4/83	1.2	8	Moore, Les	Is It My Software or My Hardware?
Kermit	6/84	2.3	40		Kermit Communications Programs
Languages	4/85	3.2	42	Salkin, Bill	One Programmer's Opinion
Legal	10/83	1.5	9	Dunaway, Robert	An Agreement Outline
Legal	12/83	1.6	33	Dunaway, Robert	Highpoints of Software Protection
Modula 2	6/87	5.3	38	Swan, Tom	Modula-2 for Pascal Lovers
OS	12/85	3.6	24	Moak, Ronald	The Santa Cruz Operation Xenix for the IBM PC
OS	2/86	4.1	39	Hardenbergh, Hal	An Operating System for the People
On Graphics	2/87	5.1	36	Abrash, Michael	Inside the EGA
On Graphics	4/87	5.2	40	Abrash, Michael	Parallel Processing with the EGA
On Graphics	6/87	5.3	30	Abrash, Michael	EGA Data Control
On Graphics	8/87	5.4	30	Abrash, Michael	EGA Set/Reset Capabilities
Pascal	2/83	1.1	7	Norton, Peter	Batch Files & Pascal
Pascal	8/83	1.4	15	Pascal S.I.G.	The Pascal SIG
Pascal	10/83	1.5	15	Morrell, Corey	Pascal Check Program
Pascal	12/83	1.6	3	Data Base Decisions	DOS Calls in Pascal
Pascal	2/84	2.1	20	Wallace, Bob	PC Write and Quicksoft

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

Department	Date	Vol.	Page	Author	Title
Pascal	4/84	2.2	46	Floyd, Chet	Tools, Tools2, and View Manager
Pascal	6/84	2.3	38	Morrell, Corey	Linking Assembler Subroutines to Pascal and BASIC
Pascal	8/84	2.4	45	Gore, Jim	ES/P: A Syntax-Checking Pascal Editor
Pascal	10/84	2.5	44	Floyd, Chet	Link up to Pascal's Pointer Power
Pascal	12/84	2.6	31	Stewart, Jay	Pascal Object Code Size Solution
Pascal	4/85	3.2	30	Swan, Tom	Scanning in Turbo Pascal
Pascal	6/85	3.3	37	Jones, Daniel J.	Confessions of a Pascal Programmer
Pascal	8/85	3.4	30	Swan, Tom	Calling MS-DOS from Turbo Pascal--It's a Breeze
Pascal	10/85	3.5	28	Minasi, Mark	Turbo Pascal and MASM: Making Them Work Together
Pascal	10/85	3.5	38	Swan, Tom	Getting Used to Turbo Pascal 3.0
Pascal	12/85	3.6	33	Rubenking, Neil	Turbo Pascal 3.0 and Subdirectories
Pascal	2/86	4.1	23	Mowat, John G.	Combining Binary Trees
Pascal	2/86	4.1	31	Kumar, Arvind	Using DOS to Call Programs from Turbo Pascal
Pascal	4/86	4.2	22	Swan, Tom	*Funky String Functions*
Pascal	4/86	4.2	26	Rubenking, Neil J.	Capturing File Handles in Turbo Pascal
Pascal	6/86	4.3	27	Lubkin, Bela	INT24.PAS: A Critical Error Handler for Turbo Pascal
Pascal	8/86	4.4	40	Swan, Tom	Teach an Old Compiler a New Trick
Pascal	12/86	4.6	32	Smith, James M.	Turbo Inline Pascal Code
Pascal	12/86	4.6	39	O'Brien/Cheung	Assembler Source Code for DOS Function Calls
Pascal	2/87	5.1	48	Swan, Tom	*How Many Holes Does It Take to Fill an IBM PC?*
Pascal	4/87	5.2	44	CompuServe Pascal SIG	Optimizing Turbo Pascal
Pascal	10/87	5.5	28	Newlin, John J.	TSRs Using Expanded Memory with Turbo Pascal
Pascal	10/87	5.5	42	Greenblatt, Jon	A Note on Optimizing Turbo Pascal
Pascal	12/87	5.6	26	Newlin, John J.	The Shell Game
Pascal	12/87	5.6	54	Swan, Tom	Turbo Pascal's Renaissance
PJ Forum	2/83	1.1	13	Detesto, Frankie	PJ Forum: Benchmarks & Wart Hogs
PJ Forum	4/83	1.2	15	Johnson, Dion L.	PJ Forum
PJ Forum	6/83	1.3	29	Bennett, Neil	I'll Be C-ing You
PJ Forum	8/83	1.4	27	Bennett, Neil	How to Write Fast Programs
PJ Forum	10/83	1.5	11	McQuaid, Robert T.	The Copy Protection War
PJ Forum	12/83	1.6	11	Barkley, Paul A.	*Structured Programming Revisited*
PJ Forum	2/84	2.1	8	FNE	DTACK Grounded--A Fable for Our Time
PJ Forum	4/84	2.2	6	Greco, Frank D.	Trying to Make an Elephant Do Break Dancing
PJ Forum	6/84	2.3	6	Greco, Frank D.	Redirection in the PC/UNIX World
PJ Forum	4/84	2.4	4	FNE	DTACK Grounded--Busted BASIC and Mass-Market UNIX
PJ Forum	6/84	2.5	4	FNE	DTACK Grounded--The 68000 and 80286
PJ Forum	8/84	2.6	8	FNE	DTACK Grounded--INTEL 80286 Bugs
PJ Forum	2/85	3.1	7	Jaeschke, Rex	C--Its Uses and Abuses
PJ Forum	2/85	3.1	28	FNE	DTACK Grounded--Surprising Memories
PJ Forum	4/85	3.2	5	FNE	DTACK Grounded--UNIX! (Again)
PJ Forum	6/85	3.3	3	Larrett, Richard	NAPLPS and the Professional Programmer
PJ Forum	8/85	3.4	5	FNE	DTACK Grounded--More UNIX Stuff
PJ Forum	10/85	3.5	3	FNE	DTACK Grounded--Windows Done Here?
PJ Forum	10/85	3.6	10	FNE	DTACK Grounded--C's HLL Overhead
PJ Forum	2/86	4.1	5	FNE	DTACK Grounded--Those PC/AT Disk Problems
PJ Forum	4/86	4.2	5	FNE	DTACK Grounded--The Intel 80386, Single- vs. Multi-user
PJ Forum	6/86	4.3	8	FNE	DTACK Grounded--RISC Paging, Software Talent, Aloha 32032
PJ Forum	8/86	4.4	6	Hardenbergh, Hal W.	DTACK Grounded--Adult Toys, Tomorrow's Micros, HALGOL Update
PJ Forum	10/86	4.5	5	Hardenbergh, Hal W.	DTACK Grounded--32-bit PCs, the 68040, and More
PJ Forum	12/86	4.6	7	Hardenbergh, Hal W.	DTACK Grounded--The Clipper; Disk Access Schemes, Word Perfect
PJ Forum	2/87	5.1	8	Hardenbergh, Hal W.	DTACK Grounded--Standards--Yaay! Booo!; Graphics Hardware
PJ Forum	4/87	5.2	10	Hardenbergh, Hal W.	DTACK Grounded--Marketing Plans and Raucous Laughter
PJ Forum	4/87	5.2	16	Duncan, Ray	Starting Your Own Software House
PJ Forum	6/87	5.3	10	Duncan, Ray	Starting Your Own Software House: Getting the Word Out
PJ Forum	8/87	5.4	12	Duncan, Ray	Starting Your Own Software House: Technical Support
PJ Forum	10/87	5.5	10	Walker, Christian	Slouching towards the 386--The Intel 80386 Processor versus IBM's Operating System 2
PJ Forum	12/87	5.6	10	Walker, Christian	The Second Coming--Multics, OS/2, Paging and Segmentation
PJ Interview	10/83	1.5	5	Salter, Brett	Marketing Your Own Software
PJ Interview	2/85	3.1	4	Moak, Ron	November's Comdex
PJ Interview	10/87	5.5	16	Greco, Frank D.	An Interview with Jack Davis, President of Metagraphics, Or, I Never Meta Man I Didn't Like
PJ Profile	2/83	1.1	5	Norton, Peter	Peter Norton and the *Norton Utilities*
PJ Profile	4/83	1.2	10	Lewis, Ted	PJ Profile
PJ Profile	6/83	1.3	20	Smolek, Ken	PJ Profile
PJ Profile	2/84	2.1	5	Jones and Rohde	Consulting
PJ Profile	4/84	2.2	4	Kernighan, Brian	C and C Plus Plus
PJ Profile	6/84	2.3	9	Floyd, Chet	Software Development Environments
PJ Profile	8/84	2.4	9	Colvin, Neil	*Writing a Custom ROM BIOS for PC DOS/MS DOS Machines*
PJ Profile	10/84	2.5	6	Hunt and Lepkoff	Devloping APX Core Executive, a Multi-tasking, Multi-windowing Environment
PJ Profile	12/84	2.6	5	Salter, Brett	Periscope--Building a Hardware Board for the IBM PC
Prolog	10/86	4.5	18	Minasi, Mark	An Introduction to Prolog
Prolog	10/86	4.5	24	Baxter, Lew	It's Not Real Prolog
Prolog	10/86	4.5	27	Swan, Tom	Turbo Prolog: Another Miracle?
Redirections	8/84	2.4	13	Greco, Frank D.	RISC, Transputors, and Mass Storage
Redirections	10/84	2.5	10	Greco, Frank D.	Who's Supporting What and Why
Redirections	12/84	2.6	10	Greco, Frank D.	AT&T Surprises All
Redirections	4/85	3.2	11	Greco, Frank D.	Artificial Intelligence
Redirections	6/85	3.3	6	Greco, Frank D.	Report from Uniforum
Redirections	8/85	3.4	13	Greco, Frank D.	C Programming Books
Redirections	10/85	3.5	10	Greco, Frank D.	C Programming Books, Part 2
Redirections	12/85	3.6	18	Greco, Frank D.	Report from the Comdex and Usenix Shows
Redirections	4/86	4.2	15	Greco, Frank D.	Industry News: PC Pact with Interactive Sytems
Redirections	6/86	4.3	18	Greco, Frank D.	Sun Microsystems' Next Generation PC
Redirections	8/86	4.4	14	Greco, Frank D.	Domestic and Overseas Report
Redirections	10/86	4.5	10	Greco, Frank D.	Redirections
Redirections	12/86	4.6	15	Greco, Frank D.	Unix Books,Part 1
Redirections	12/86	4.6	21	Budlong, M. J.	How Many Amoebas at One O'clock?
Redirections	2/87	5.1	14	Greco, Frank D.	286 DOS Cometh

PRODUCED BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

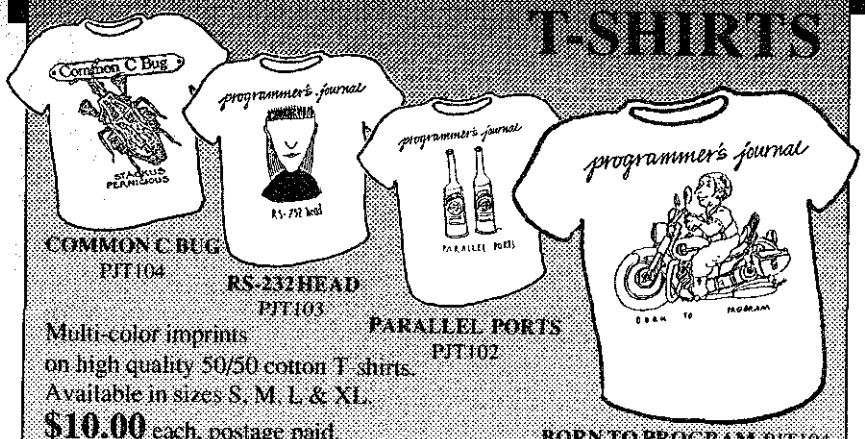
Department	Date	Vol.	Page	Author	Title
Redirections	4/87	5.2	22	Greco, Frank D.	Pink Noise
Redirections	6/87	5.3	16	Greco, Frank D.	Unix Books, Part 2
Redirections	8/87	5.4	18	Greco, Frank D.	Unix Books, Part 3
Redirections	12/87	5.6	14	Greco, Frank D.	Report from the Usenix Conference and Pink Noise
Review	6/83	1.3	6		Text Editors
Review	8/83	1.4	34	Bumpas, Jim	Hostcomm
Review	8/83	1.4	34	Moak, Ron	The Programmer
Review	8/83	1.4	34	Whitman, David	The Randomhouse Proofreader
Review	10/83	1.5	37	Moak, Ron	BASIC Developement System
Review	10/83	1.5	35	Kaplon, Morton F.	EDIX/WORDIX
Review	12/83	1.6	43	Bohle, Jim	The dBase II Window
Review	2/84	2.1	42	Wanderman, Richard	Punctuation and Style
Review	12/84	2.6	43	Lichty, Tom	Thinktank
Review	2/85	3.1	34	Salkin, Bill	Periscope, a Symbolic Debugger
Review	2/85	3.1	39	Volkstorf, J. Edward, Jr.	Four Programming Books: *Writing Efficient Programs, Bluebook of Assembly Routines for the IBM PC & XT, Computer Game-Playing Theory and Practice, C Programmer's Library*
Review	4/85	3.2	45	Jaeschke, Rex	Blaise's Ctools
Review	6/85	3.3	45	Volkstorf, J. Edward, Jr.	*Systems in Organizations, Bugs + Features*
Review	8/85	3.4	39	Salkin, Bill	The Set Tools
Review	10/85	3.5	42	Moak, Ronald	Topview Programmer's Toolkit
Review	12/85	3.6	45	Barkley, Paul	Clipper Sails on dBASE Seas!
Review	12/85	3.6	49	Jaeschke, Rex	Blaise's Ctools2
Review	2/86	4.1	46	Volkstorf, J. Edward, Jr.	Two Microcomputer Graphics Books--*MicroComputer Displays, Graphics and Animation, GKS Primer*
Review	4/86	4.2	50	Hoyt, Michael	Programming for Printer Output
Review	6/86	4.3	51	Keller, Robert	PJ Tries Out Some Time-saving 80286 Coprocessor Boards
Review	10/86	4.5	52	Keller, Robert	Short Boards
Review	12/86	4.6	46	Barkley, Paul	The Cannon A1 Laser Printer
Review	12/86	4.6	54	Barkley, Paul	Genifer: A Powerful dBASE Applications Generator
Review	12/86	4.6	58	Feldmann, Peter	Advanced Trace86: A Debugger and More
Review	4/87	5.2	62	Abrash, Michael	Turbo C: A First Look
Shareware	12/87	5.6	52	Newlin, John J.	Shareware Professionals Form Association
Style	2/83	1.1	3	Lewis, Ted	A Style Manual for Software Authors
System Analy	2/83	1.1	20	Huntington, Bruce	System Analysis
Tools	2/83	1.1	14	Estes, Greg	Tools of the Trade
Tools	4/83	1.2	17	Estes, Greg	Tools
Tools	6/83	1.3	26	Estes, Greg	Tools
Tools	8/83	1.4	25	Estes, Greg	Default Values and Program Variables
Tools	10/83	1.5	29	Moak, Ronald	Programmer's Calculator
Tools	6/84	2.3	32	Buchanon, Wat	Hybridizing Assembly and BASIC Compiler Routines
Tools	8/84	2.4	30	Brown, Robert E.	Using the Pmate Macro Language
Tools	10/84	2.5	38	Gore, Jim	Rtool, a PC-resident Programmer's Toolkit
Tools	12/84	2.6	33	Jones, Daniel J.	Correcting Oversights in debug.com
Tools	2/85	3.1	22	Baer, Dennis	Structured Programming Language
Tools	4/85	3.2	38	Foley, Patrick	Calculate Record Layouts, Automatically!!
Tools	6/85	3.3	40	Cox, Terry W.	Tools for Better Programs
Utilities	6/83	1.3	23	Salter, Brett	Peeks and Pokes
Utilities	8/83	1.4	24	Larratt, Richard	Disk Explorer
Utilities	10/83	1.5	34	Hailey, Bill	Prokey
Utilities	12/83	1.6	42		Polylibrarian/Object Module Maintenance
Utilities	2/84	2.1	41	Estes, Greg	Ultra Utilities
Utilities	8/85	3.4	33	Robinson, Bernard H., Jr.	DOS 2.0 Sort and Find Filters
Utilities	4/86	4.2	38	Baker, Steven	Memory Resident Control over Print Screen Functions
Utilities	8/86	4.4	26	Hoyt, Michael	Programming the WordStar Document File Conversion Utility: Again and Again and Again...
Utilities	8/87	5.4	44	Bronstein, Daniel A.	MWWIND--A Menu with Windows Utility
VODS	2/84	2.1	27	Wallach, Charles	VODS--VDT/VODS Bulletin ♦

BACK ISSUES

Great code and irreverent humor from *PJ*'s first five years. 10% discount on orders of 10 or more issues. Order yours today—supplies are limited.

$7.00 each, postage paid.

T-SHIRTS



COMMON C BUG PJT104

RS-232 HEAD PJT103

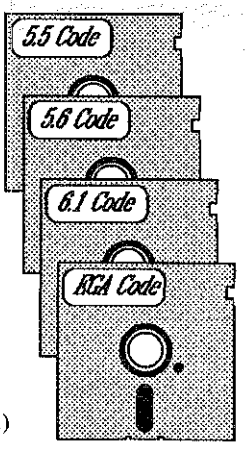
PARALLEL PORTS PJT102

BORN TO PROGRAM PJT104

Multi-color imprints on high quality 50/50 cotton T shirts. Available in sizes S, M, L & XL.

$10.00 each, postage paid.

CODE DISKS



The code from issues 5.5, 5.6, 6.1 & Michael Abrash's EGA code is now available on disk.

$10.00 each postage paid.

(See page 82 for access via BBS.)

PRODUCED 2024 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

Simply the BEST C and Pascal on AT, 386, Sun, Apollo, RT, VAX, 370

"The most rock-solid C compiler in the industry. Superb technical support and portability. Superior code generated."
Gordon Eubanks, Symantec — Q&A (386).

"It simply works, with no trouble, no chasing strange bugs, and excellent warning and error messages ... a professional product."
Robert Lerche, Bay Partners.

"For large-scale software development, the highest quality C compiler available on the market today. Pragmas are great. Quality of support is exceptional." **Randy Neilsen, Ansa—Paradox** (DOS,OS/2).

"15% smaller and 15% faster than Lattice C."
Robert Wenig, Autodesk.

"Our software is running anywhere from 30 to 50% faster than when compiled under Lattice." **David Marcus, Micronetics.**

"We switched from Lattice due to a 10% reduction in code size. The compiler is very stable." **Lee Lorenzen, Ventura Software — Ventura Publisher, marketted by Xerox Corp.**

"Best quality emitted code by any compiler I've encountered. Often amazing." **Bill Ferguson, Fox Software — FoxBase (386).**

"Messages sometimes pointed out type mismatches, incorrect-length argument lists, and uninitialized variables that had been undetected for years [in 4.x bsd]." **Larry Breed, IBM ACIS [RT PC].**

"Diagnostics turned up bugs missed by other compilers. Rapid bug fixes by technical support, someone who knew what he was talking about. 80386 code is well optimized."
Tim Addison, Logistics Data Systems.

"386 protected mode support is fantastic, especially the access to large amounts of memory. It's mainframe compute power on a PC." **Dan Eggleston, Viewlogic.**

"The preprocessor supplied with Professional Pascal is quite useful. The code quality and control over segmentation and memory models are superior to MS Pascal." **Bob Wallace, QuickSoft.**

Check Out These Reviews

- **High C ™:**

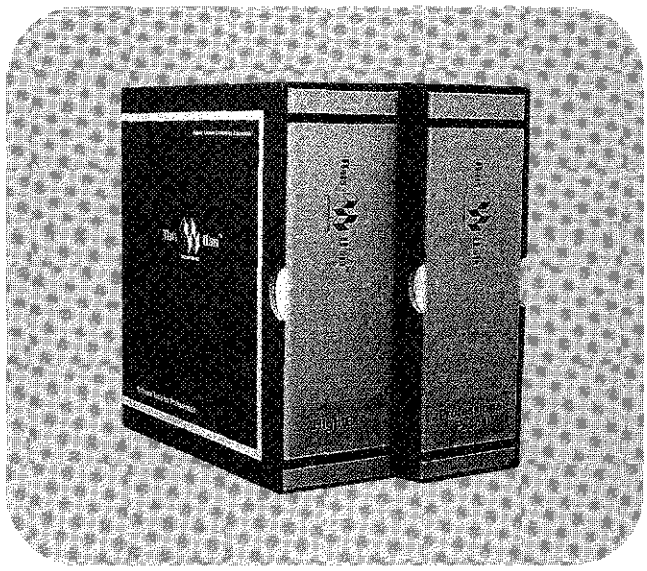
Computer Language	February 1986, '87	
Dr. Dobb's Journal	August 1986	
PC Magazine	Jan. 27, 1987	(80386 version)
Dr. Dobb's Journal	July 1987	(80386 version)
BYTE Magazine	November 1987	(80386 version)

- **Professional Pascal ™:**

PC Magazine	Dec. 29, 1985	
Computer Language	May 1986	
PC Tech Journal	July 1986	
Journal of Pascal, Ada, & Modula-2	Nov.-Dec. 1986	
BYTE Magazine	Dec. '86, June '87	(80386 version)

Why MetaWare compilers

- They are specifically designed for serious software developers.
- They are reliable and robust: they don't break at every turn.
- Their generated code is the best, or near best, on each architecture.
- Their superior diagnostic messages help you produce better products more quickly.
- Your source can be ported with ease to the most popular systems.
- You can link mixed-language modules from our compilers, others
- You can benefit from high-level, personal technical support.
- You can take advantage of the latest ANSI C extensions, and/or extensive Pascal extensions. **High C** has been tracking the ANSI Standard for two years; **Professional Pascal** will soon have a VS Pascal compatibility switch and several Apollo Pascal ext'ns.
- You can take advantage of the latest 387 and Weitek 1167 support — we have the only compilers with Weitek real mode support.



Power Tools for Power Users

Ashton-Tate: dBase III Plus, MultiMate; Autodesk: AUTOCAD, AUTOSKETCH (8087, '387, Weitek); Boeing Computer Services (Sun); CASE Technology (Sun); CAD/CAM giant Daisy ('86, '386, VAX); Deloitte Haskins & Sells; Digital Research: FlexOS; GE; IBM: 4.3/RT, 4680 OS; Lifetree Software (Pascal): Volkswriter Deluxe, GEM-Write; Lugaru: Epsilon; NYU: Ada-Ed cmplr; Semantec: Q&A; Sky Computers; ... (Product names are trademarks of the companies indicated.)

Industrial-Strength

MetaWare C and Pascal compilers are designed for professional software developers. These tools are loaded with options to control them for special purposes. You can adjust the space-time trade-off *in code quality. You can adjust external naming conventions* to agree with linkers and operating systems. You can specify segment names for segmented architectures, and to help place code or data in particular places for embedded applications. You can select from five memory models for the 8086 family. And on and on.

A Partial List of Optimizations

Common subexpression and dead-code elimination, retention and reuse of register contents, jump-instruction size minimization, tail merging (cross jumping), constant folding, short-circuit evaluation of Boolean expressions, strength reductions, fast procedure calls, automatic mapping of variables to registers (where advantageous), ...

"Platform" — Code Quality

Sun,Apollo,SGI	— 18%, 3%, 26% > resident compiler (Dhrystone).
PC: DOS, OS/2	— 3-10% > Microsoft C; 30% > MS Pascal, LatticeC.
386 32-bit DOS	— no competitors, since November, 1986.
286, 386 UNIX	— 66% better than pcc (Dhrystone, 386).
VAX VMS	— ≈ DEC's excellent C and Pascal; better features.
VAX Ultrix	— 19% > pcc (Dhrystone); much > Berkeley Pascal.
RT PC/4.3bsd	— 89% > the original port of pcc (Dhrystone).
370 CMS,UNIX	— much better than any C, and VS Pascal.
AMD 29000	— >40,000 Dhrystones! Available in Q2, cross.

(408) 429-6382, telex 493-0879. ***Since 1979.***

MetaWare™
INCORPORATED

903 Pacific Avenue, Suite 201 • Santa Cruz, CA 95060-4429

The Clear Choice for Large Programming Projects — PC Tech J.

© 1987 MetaWare Incorporated. MetaWare, High C, Professional Pascal, and DOS Helper are trademarks of MetaWare Incorporated. Others and their owners are duely respected.

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

80386 Development Tools

We are about to witness the "clash of the Titans" in the 386 universe. The major minicomputer compiler vendors have been busy porting their 32-bit Unix compatible compilers to the 80386. These minicomputer vendors traditionally provide highly optimizing compilers for 32-bit machines and sell to the Unix community better products than the compilers that come with Unix. These products have been marketed on performance, not cost (C, FORTRAN and other compilers come free with Unix). The rivalry will be between the PC compiler vendors and these workstation suppliers inspired by the 386 to move into developing compilers for the PC. We all will benefit from the competition.

386|ASM/LINK
Phar Lap Software, Inc.
60 Aberdeen Avenue
Cambridge, MA 02138
617/661-1510
DOS Assembler $495 Debugger $195

Phar Lap Software initiated the competition with their 80386 Software Development Series. They developed an extension to the Intel Object Module Format (OMF-86) for the 80386 called Easy OMF-386. Phar Lap has been selling a 386 assembler, 386 linker, 386 librarian, monitor program (MINIBUG), and DOS extender (RUN386)—all based on this Easy OMF—to test and run 386 programs in the protected mode. Recently, they've added a (full) symbolic debugger (386DEBUG), which is sold separately. The Phar Lap Easy OMF-386 has become a defacto industry standard for 80386 vendors. Intel has devised new OMFs for their new CPUs, calling them OMF-286 and OMF-386, but these new OMFs are completely different from their earlier OMF-86 and have not been adopted outside Intel. In a strange turn of events, Intel has so far considered their own OMF-386 to be a proprietary format and has not published a technical description for users. And we all thought they were selling chips?

Phar Lap's development tools were written in C and are available under DOS or on a VAX. The DOS tools (except the debugger and RUN386) do not require an 80836 and run on a PC with 512K. The assembler supports Intel syntax for the 8086/87 through 80386/x87 chips and supports the full 32-bit registers of the 80386 using both the protected and real mode instructions. Included in the package are examples and object files to support the Microsoft C, RM/FORTRAN, and other compilers using the 32-bit registers in an otherwise 16-bit mode. In the DOS world, all of the 386 vendors currently depend on Phar Lap for their assembler and linker. Phar Lap has recently developed LinkLoc, a ROM tools development package for embedded systems ($295). LinkLoc supports Intel's OMF-286 and Intel's undocumented OMF-386 formats. Phar Lap is constantly improving these products and issues monthly technical newsletters and updates to registered users.

386|DOS Extender (RUN386)
Phar Lap Software, Inc.
60 Aberdeen Avenue
Cambridge, MA 02138
617/661-1510
Runtime License $995

Once someone has developed a true 80386 application, Phar Lap includes in its assembler package RUN386, their 386 DOS extender. RUN386 is a control program that sets up the 80386 in the protected mode, loads a 32-bit program into memory, and executes it. Two other vendors (IGC and A.I. Architects) now supply a similar product. All of these 386 DOS extenders set up the equivalent of DOS function calls using software interrupts that operate in the protected mode using 32-bit registers and then pass the requests on to the real MS-DOS operating system in low memory. On 386 machines, this allows the execution of very large programs no longer limited to the 640K of DOS and using full 32-bit registers. Phar Lap's RUN386 actually switches back to the real mode for DOS calls from the protected mode. On the 386, this switch is quick (100 microseconds) since it's supported by software (80386 instruction), not the hardware kludge used on the IBM AT (80286). The Phar Lap runtime is a flat memory model with several predefined 80386 segment selectors for memory mapping. Their DOS Extender uses about 100K of low memory aside from DOS, leaving about 500K of low memory plus any extended memory for protected mode use. The Weitek math coprocessor is supported.

The protected mode DOS calls are similar to the normal DOS calls except that 32-bit registers are used. Phar Lap has added a few extensions to the standard DOS calls (get version returns a DOS extender signature in the high word of the EAX register). The latest versions of RUN386 also incorporate support for Quarterdeck's DESQview and the Compaq extended memory manager. Applications developed with Phar Lap's DOS Extender will be able to multitask under DESQview for the 386. Phar Lap also distributes *386 Control Program Compatibility Guidelines*, developed jointly with Quarterdeck. The guidelines are an attempt to avoid conflicts, like those that occurred with TSRs under DOS, among 386 control programs.

For distribution purposes, Phar Lap sells an unlimited runtime license and a BIND386 and CFIG386 utility. The binder is used to bind the 386 system kernel to the front end of an application, turning it into a DOS EXE file. The configuration utility allows customizing of the runtime memory environment. Paradox-386 uses the Phar Lap DOS Extender.

X-AM Development Environment
Intelligent Graphics Corporation
4800 Great America Parkway
Santa Clara, CA 95054
408/986-8373
$595

IGC started out marketing graphics hardware and software for the PC. Last May, they acquired the 80386 development team working on virtual machine software for the 80836 from Softguard Systems. X-AM (eXtended Address Mode) is an 80386 development environment similar to Phar Lap's DOS Extender environment. X-AM includes VM/RUN (the 80386 runtime environment) and VM/DEBUG (a symbolic debugger much like SYMDEB). VM/RUN uses technology similar to IGC's VM/386 advanced multitasking control program. The VM in these product logos stands for Virtual Machine—the development team was recruited from the IBM mainframe VM operating system environment.

X-AM differs from Phar Lap's and A.I.'s development environments (A.I.'s OS/286 and OS/386 Developer's Kit is discussed below) in that the 80386 is switched from protected mode (32-bit) to virtual 8086 mode for DOS calls. X-AM sets up a flat memory model using paging with memory for protected mode applications mapped above 32 megabytes. The flat memory model is similar to the Motorola 68000 or IBM/ 370. No segment registers are used. For example, color video memory would be mapped at B8000H, and BIOS data areas at 400H—their actual physical address. VM/RUN uses a minimum of 200K of the lower 640K memory for the VM runtime monitor, but increased buffer areas, space for TSRs, and EXECing to other programs can eat up all of the lower 640K of physical memory. VM/RUN also supports the Weitek chip. VM/RUN does not currently implement virtual memory—an application uses only physical memory.

X-AM also includes the Make Profile (MP) utility to define any special loading characteristics for VM/RUN. The X-AM package currently requires use of the Phar Lap linker to generate a REX file (their extension for protected mode relocatable executable files). IGC also sells the Phar Lap Assembler/Linker and MetaWare 80386 C

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

That's right. 4 times faster.

Clocking in at over 75,000 lines per minute on a 6MHz IBM AT, OPTASM is four times faster than Microsoft's MASM 5.0. **4 times faster** — that's 400% more throughput!

But speed is only one part of it. OPTASM is nearly 100% compatible with MASM 5.0 (except 386 support).

It is the only single assembler capable of supporting the various incompatibilities between MASM 3, 4 & 5. That makes OPTASM **more MASM compatible than any single version of MASM!**

Other features? OPTASM generates smaller code without ever generating extra NOP's. It automatically handles jumps out of range, up to 15,000 symbols and most of MASM's phase errors. It also boasts a built in MAKE and simplifies segmentation.

That's why we can make our OPTASM challenge: Test OPTASM head to head against MICROSOFT MASM 5.0. Order **both** assemblers with their 30-day guarantees. In a lot less than 30 days, you'll see just how dazzling OPTASM's speed really is. You'll realize that we're compatible, easier to use, and deliver many more important features than MASM. So accept our challenge. Try both assemblers. **Four times faster and more features, too.** We know which one you'll send back.

Write or call us to order or for our detailed brochure.

SLR SYSTEMS

1622 N. Main Street
Butler, PA 16001
412-282-0864
BBS 412-282-2799
Telex 559215
800-833-3061

OPTASM: $195 Guaranteed returnable within 30 days.

Number **165** on the Reader Service Card.

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

and Pascal compilers. The price of X-AM includes a runtime license allowing bundling of the VM/RUN modules with copies of any applications developed. FoxBASE+/386 uses the IGC X-AM environment.

OS/286 and OS/386 Developer's Kit
A.I. Architects, Inc.
One Kendall Square
Cambridge, MA 02139
617/577-8052
$495 each

A.I. Architects manufactures a series of high performance 80386 coprocessor boards for the PC that have been used in LISP and AI applications. They also sell a Developer's Kit that includes the OS/286 or OS/386 kernel, a symbolic debugger, and an EXPRESS utility to generate a protected mode executable file. Their OS/x86 kernel differs substantially from the other vendors'. OS/286 is a DOS extender for 80286 machines. It supports a segmented memory model on the 80286, allowing applications to break the 640K memory barrier under DOS. Lahey's F77L-EM FORTRAN compiler uses this OS kernel. This same memory map is also supported under OS/386 on the 80386 along with the more common flat memory model. The executable files are named with the extension EXP (EXecutable Protected mode), the same extension adopted by Phar Lap for their default executable files even though the Phar Lap and A.I. files have slightly different formats and the 386 DOS interfaces may differ slightly. A.I. calls Phar Lap's files PLX (Phar Lap eXtended) and can run them as well as their native format. The strength of the A.I. product is in its support for larger applications under DOS on the 80286. A number of special function calls are supported for contiguous memory allocation for larger data structures. The 86 segments are now really segment selectors.

On the 80386, OS/386 switches from protected mode to real mode for DOS calls. OS/386 uses about 40K of memory in the lower 640K. But most protected mode programs by default will use only memory above 1 megabyte unless special OS/386 calls are used. Language support is provided for Microsoft C and FORTRAN and for Lattice C. Special versions of the OS kernels are available for A.I.'s Hummingboard 80386 coprocessor boards.

Now the 386 Compilers

High C and Professional Pascal
MetaWare, Inc.
903 Pacific Avenue, Suite 201
Santa Cruz, CA 95060
408/429-6382
386 DOS $895 8086/286 MS-DOS $595

MetaWare was the first vendor to supply compilers under DOS for the 80386. Both their Pascal and their C compilers have been available for the 80386 for about a year. The two compilers run under DOS 2.x or higher and generate optimized object code that can be linked using the Phar Lap 386|LINK linker. The resulting program can be run in the protected mode on the 80386 and has full access to the large 32-bit address space and the full 32-bit registers of the 80386. The resulting executable file will run on DOS 3.x or higher under any of the 386 DOS extenders—IGC's X-AM, Phar Lap's DOS Extender, or A.I. Architects' OS/386. The MetaWare compilers are highly optimizing and offer excellent code generation. Support is provided for embedded applications. The compilers and runtime libraries support both the 80287 and 80387 math coprocessors. MetaWare has recently added support for the Weitek 1167 math coprocessor.

The two compilers are functionally the same as the high-quality High C and Professional Pascal 8086/286 compilers. MetaWare also sells the 80386 developer kits from Phar Lap, IGC, and A.I. Architects. They have also recently added a DOS Helper, a set of powerful Unix-like tools to enhance MS-DOS for $49.95—included free with the MetaWare compilers. DOS Helper includes FIND, TAIL, MV, LS, CAT, UNIQ, FGREP, and WC utilities. MetaWare supplies a family of native and cross compilers that generate code for the 8086/286/386, VAX, 68000/010/020, 32000, RT PC, and 370, and that run on various operating systems hosted on those processors.

SemWare Announces QEdit Advanced the "quick editor"

QEdit is one of the smallest, fastest, multi-file editors available for IBM compatible computers and has many features to enhance your productivity.

- Lightning fast, especially in file loading and saving, screen display, cursor movement, and global searching and replacing.
- Edit multiple files, as many and as large as available memory allows. QEdit can use all available memory, up to 640KB. Ideal for multi-file languages and all those .PRG files.
- Windowing allows up to 7 files or 7 views of the same file to be seen at once.
- You do not have to "learn" another editor with QEdit. QEdit's commands can be configured to any keys you like.
- Screen colors, tab settings, default file type and many other options can be configured the way YOU like it.
- Up to 99 named scratch buffers may be created and edited. Commands are available for easy cut and paste operations.
- Invoke a DOS "shell" or execute DOS commands directly from QEdit.
- Both line and stream block marking are supported.
- Keyboard macros allow automation of commonly used command sequences or text entry.
- Commands to shift an entire block left or right, to adjust indentation levels when using structured languages.
- Garbage stack provides restoration of the last "n" (user-settable) deletions.

Introductory Price
49.95
Mastercharge & Visa Accepted
Add $3.00 Shipping
Georgia Residents please include 5% sales tax

QEdit Advanced requires an IBM PC/XT/AT or Compatible, IBM PS/2 30 - 80, 128KB of available memory. Please specify disk format required.

To Order: Heritage Papers, Danielsville, GA 30633
Phone Orders: (404) 789-3226

Number **148** on the Reader Service Card

LPI FORTRAN, COBOL, BASIC,
Language Processors, Inc.
400-1 Totten Pond Road
Waltham, MA 02154
617/890-1155
$695 DOS

Because C and Pascal have been available from MetaWare, most minicomputer vendors have focused on FORTRAN and other languages for their first releases onto the 80386. Language Processors has been porting a whole family of high performance compilers to the 80386 environment—BASIC, C, COBOL, FORTRAN, Pascal, PL/I, RPG II, and LPI-DEBUG, their interactive source code debugger. These products will be available for two environments: (1) Unix and Xenix; and (2) Phar Lap's 386 Dos Extender. All of these products will be available using SCO Xenix or Unix System V (testing under Microport). Under DOS, the C and RPG II compilers will not be available. The DOS compilers include the Phar Lap assembler, linker, librarian, and runtime. LPI currently supports the 80287/80387 chips and are working on adding Weitek math chip support. The compilers generate object files directly, and are designed to run under DOS 3.x. LPI FORTRAN has already been validated by the National Bureau of Standards (NBS) running Xenix on the Intel 386. A runtime only license is required for some

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

of the language products (not FORTRAN) and costs from $70 to $200 depending on the language, and annual updates vary from $115 (FORTRAN) to $200 (COBOL). The Debugger costs an additional $695.

LPI has been supplying similar products to Unix vendors including Altos, AT&T, Arete, Convergent Technologies, Honeywell, NCR, Prime, Sun, and Unisys on various hardware platforms. Their FORTRAN, COBOL, and BASIC compilers have been validated by the NBS on a variety of Unix hardware. SPSS is using LPI FORTRAN to port its statistics package to the 80386.

NDP FORTAN-386, C, and Pascal
MicroWay
P.O. Box 79
Kingston, MA 02364
617/746-7341
$595

MicroWay has been working with Green Hills Software (Glendale, California), a major minicomputer compiler vendor, to offer their family of compilers under DOS for the 80386 using the Phar Lap DOS Extender. MicroWay has traditionally specialized in numeric coprocessor hardware and supporting software on the PC. This is their first compiler effort, and they have been working with Green Hills to add Weitek math coprocessor support under DOS. Green Hills Software makes advanced optimizing compilers for FORTRAN 77, C, and Pascal that run on Unix System V, Berkeley Unix, and now 386/DOS. These compilers are globally optimizing, providing some of the fastest code execution in the business. The compilers support the 80287, 80387, and Weitek math coprocessors. Be forewarned that these compilers require lots of memory for compilation because of the information collected during the optimization process. Under the 386 DOS Extender, 3 or 4 megabytes should be considered a minimum configuration. The compilers generate assembly language files for the Phar Lap assembler and linker.

The Phar Lap development tools must be purchased in addition to the compiler. Green Hills Software also offers cross-compilers hosted on VAX/VMS, VAX/Unix, and 386/DOS. Green Hills has code generators for the Intel 80386, the Motorola 68000/10/20, the Fairchild Clipper, the National 32000, and the Weitek XL RISC family.

SVS FORTRAN
Science Applications International Corp.
5150 El Camino Real, Suite C-31
Los Altos, CA 94022
415/960-3322
$895

SAIC has negotiated with Silicon Valley Software (SVS) of Cupertino, California, to sell the SVS FORTRAN 77 compiler under DOS using the IGC X-AM 386 DOS extender. SAIC is a very large firm with numerous offices worldwide involved in many different consulting activities. The Los Altos office also writes a large amount of FORTRAN code. I have been told that SAIC may also market the SVS C and Pascal compilers for the 386. SVS is another minicomputer compiler vendor traditionally providing highly optimizing compilers to the Unix world (see the review of the Definicon board in this issue). The compiler generates an intermediate file that is converted to an object file for the target environment. The compiler supports the 80287/387 and Weitek coprocessors. The SVS FORTRAN compiler for the 80386 includes the SVS source code symbolic debugger, the X-AM Developer's Environment, and the Phar Lap 386|LINK linker.

1167 Software Developer's Kit
Weitek Corporation
1060 E. Arques
Sunnyvale, CA 94086
408/738-8400
free

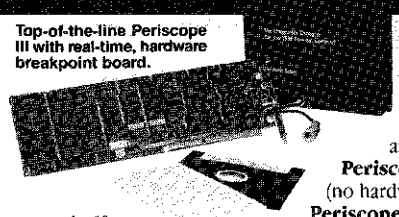
Weitek makes high performance processor chips used by a variety of hardware vendors. They manufacture numeric coprocessors, processors, and building blocks that perform from 1 MFLOPS (Millions of FLOating Point instructions per Second) to 40 MFLOPS in the high end building blocks. Intel and Weitek worked together to provide a high performance (by PC standards) numerics coprocessor for the 80386 called the Weitek 1167. The 1167 is actually a three chip set—1163 controller chip (designed specifically for 80386 interface), 1164 adder, and 1165 multiplier. This three chip set is designed to be used on a daughter board that would plug into a 121 pin interface (a superset of the 80387 chip interface). Last issue I mentioned that MicroWay has begun distribution of these coprocessor cards, which also have space for the 80387. Compaq also announced a board.

Well, what's the big deal about this Weitek coprocessor anyway? For high end applications, the Weitek 1167 promises to at least double the speed of number crunching applications (graphics, CAD/CAM, statistics, simulation) compared to an 80387 for a bit more than the cost of an 80387. This amounts to the horsepower of a VAX 8600 on your desk. Weitek estimates that 25% or more of the 2 million 80386 based machines shipped through 1991 will add a Weitek coprocessor. In the longer term, though, the potential is for much faster (5X) Weitek coprocessors for the 80386 now that a standard hardware and software interface has been defined. Keep in mind that the 1167 set is at the bottom of their product line in terms of speed.

The software interface to the 1167 is a 64K memory mapped window usually

PERISCOPE™ POWER

...Keeps you going full steam ahead when other debuggers let you down. With four models to pick from, you'll find a Periscope that has just the power you need.



. . . Start with the model that fits your current needs. If you need more horsepower, upgrade for the difference in price plus $10! And don't worry about having a lot more to learn . . . Even when you move to the most powerful model, Periscope III, an extra dozen commands are all that's involved.

Periscope's software is solid, comprehensive, and flexible. It helps you debug just about any kind of program you can write . . . thoroughly and efficiently. Periscope's hardware adds the power to solve the really tough debugging problems.

Periscope requires an IBM PC, XT, AT, or close compatible (Periscope III requires hardware as well as software compatibility); DOS 2.0 or later; 64K available memory; one disk drive; an 80-column monitor.

Periscope I includes a half-length board with 56K of write-protected RAM; break-out switch; software and manual for $345.

Periscope II includes break-out switch; software and manual for $175.

Periscope II-X includes software and manual (no hardware) for $145.

Periscope III includes a full-length board with 64K of write-protected RAM, hardware breakpoints and real-time trace buffer; break-out switch; software and manual. Periscope III for machines running up to 8 MHz is $995; for machines running up to 10 MHz, $1095.

Call Toll-Free for free information or to order your Periscope today!

MAJOR CREDIT CARDS ACCEPTED. ***800-722-7006***

The PERISCOPE Company, Inc.
1197 PEACHTREE ST.
PLAZA LEVEL
ATLANTA, GA 30361
404/875-8080

Number 6 on the Reader Service Card

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

placed below the highest addressable range on the 80386. The Weitek chip set contains 32 single precision registers, which may be grouped into 16 double precision registers. The interface has been optimized for memory access speed—writing to a location in this memory map is the equivalent of an opcode. Saving the coprocessor state is just a fast block move of memory. The coprocessor instructions can be handled as a series of macros. The Weitek Software Developer's Kit includes data sheets, software programmer's guides, and a diskette with some demo software for the 1167. Weitek also provides opcode macros and a library of math routines for the 1167.

Windowing and Graphics

Turbo C Version 1.5
Borland International
4585 Scotts Valley Drive
Scotts Valley, CA 95066
800/543-7543
$99.95 ($33.50 upgrade)

Borland has released an updated version of Turbo C. This new release includes a large set of low and high level graphics routines, full function online help, a librarian, and a fast grep utility. This new Borland Graphics Interface (BGI) supports both text and dot addressable graphics modes. The same functions are also supported in Turbo Pascal 4.0. Graphics devices supported include MDA, Hercules, CGA, EGA, VGA, AT&T 6300, 3270 PC, and the IBM 8514 display. The assembly language drivers support up to 256 colors at 1024 x 768 resolution. Graphic fonts are provided for Gothic, Sans Serif, Triplex, stroke fonts, and other fonts loadable at runtime. High level routines include functions such as pieslice() (used to draw pie charts). A full help system has been incorporated. Help for any C keyword or library call is now only a function key away. A number of bugs in Version 1.0 have been corrected, and small changes to match the ANSI C standard have been added. The Version 1.0 manuals come with a 200-page addendum.

Standards

FORTRAN 8X Standard
Global Engineering Documents
2525 Hickory Street
Santa Ana, CA 92707
800/854-7179
$50

Beginning last October and extending through February 23, the FORTRAN Standard Committee, X3J3, has submitted for public review the next FORTRAN standard, currently referred to as 8X. During this period, the committee is soliciting comments in writing, both critical and supportive. The committee then reviews and responds to every comment. If public response on an issue reaches sufficient proportions, the standard will be changed. Tom Lahey of Lahey Computer Systems told me that "the 77 standard went through meaningful changes towards the end. IF/THEN/ELSE was added, and the extended range of DO loops and stream I/O were deleted." Tom also expressed concern about some aspects of the new standard. Make your comments known.

Books

Mastering Turbo Pascal Files
by Tom Swan
Howard W. Sams & Company
4300 West 62nd Street
Indianapolis, IN 46268
317/298-5400
$18.95

Our Contributing Editor Tom Swan, who's been writing about computer programming for nearly ten years, recently authored a new book on Turbo Pascal. Tom's latest is an advanced guidebook and reference source for programmers on writing file processing programs in Turbo Pascal. *Mastering Turbo Pascal Files* covers file structure fundamentals, sequential and random access techniques, text files, DOS directories, and low-level disk I/O programming techniques. Tom also discusses searching and sorting techniques for disk data and multiuser file and record locking considerations. Tom uses numerous example programs and procedures to illustrate these concepts. As a convenience, programs in the book are available in source code from the author for $30 (order form on last page) at Swan Software in Lititz, PA.◆

"If you need source code, make sure your wallet is wide open or get VITAMIN C."
- Computer Language, June, 1987

Vitamin C

It's good for your system!

- Professional C function library
- 30 day money back guarantee
- Multiple bullet proof windows
- Easy full screen data entry
- Unlimited data validation
- Context sensitive help manager
- Menus like Lotus and Mac
- Programmable keyboard handler
- Text editor routines
- No royalties or runtime fees
- Library source included FREE
- Free technical support
- Free BBS at (214)418-0059
- Supports all major compilers including Microsoft 5.0
- VCScreen code generator too!
- UNIX version avaialable

Fast, flexible, versatile, reliable. Vitamin C delivers the vital combination that software professionals demand. Highly efficient, professionally crafted C code provides lightning fast displays required by today's window intensive programs.

High level functions provide maximum productivity and require little supporting code. Plus, Vitamin C's versatile, open ended design is full of hooks so you can intercept and plug-in special handlers to customize or add features to most routines.

VCScreen, our screen painter / code generator speeds your development even more! Simply draw your input forms using our interactive design editor and generate perfect C source code ready to compile and link with the Vitamin C library.

Vitamin C $225
Includes all source code FREE! For IBM PC, XT, AT, PS/2 and true compatibles. Specify compiler when ordering.

VCScreen $99.95
For IBM PC, XT, AT, PS/2 and true compatibles. Requires Vitamin C.

ORDER NOW!

We ship UPS. Please include $3 for ground, $6 for 2-day air, $20 for overnight, or $30 if outside the U.S. Texas residents add 7 1/4 % sales tax. All funds MUST be in U.S. dollars drawn on a U.S. bank. Visa & MasterCard accepted.

(214)416-6447

creative PROGRAMMING
Box 112097 • Carrollton, Texas 75011

Windows • Data Entry • Menus • Help • Text Editing
Plus... All Source Code FREE!

Contact this Advertiser directly

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

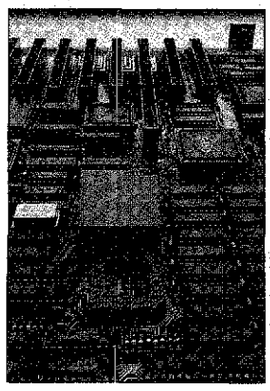
SAVE YOUR PC FROM EARLY RETIREMENT.

GET HAUPPAUGE'S NEW 386 MOTHERBOARD.

386 SPEED—ONLY $1,495

Give your PC a new lease on life! With our industry first 386 MotherBoard, your PC, PC/XT or compatible will revel in speeds equal to the Compaq DeskPRO 386. And faster. Because we've built in 1 Megabyte of high speed RAM and a 387 math coprocessor socket for speeds that will knock you off your rocker.

To keep retirement at bay, our 386 MotherBoard is compatible with the PC/AT (BIOS and I/O) — allowing you to run the new generation of DOS, OS/2. We've also included a 16-bit expansion slot that accommodates the latest I/O expansion card. No accelerator card can give you so much versatility.



With 386 power and true AT software compatibility, your business, desktop publishing and engineering applications will get a boost to boast about! **Technical Features** ▪ 16 MHz 80386 ▪ 1 Megabyte of 100 nsec 4-way interleaved RAM ▪ PC/AT compatible I/O and BIOS for support of OS/2 ▪ Seven 8-bit expansion slots ▪ Two 16-bit expansion slots ▪ One 32-bit RAM expansion slot ▪ Optional 16 MHz 80387 math coprocessor ($695)

Put the power of the 386 into your IBM PC for 1/4 the cost of a 386 computer. And put off your PC's retirement. For more information on our easy-to-install Motherboard, call **1 (800) 443-6284.** In New York, call **(516) 360-3827.**

Hauppauge Computer Works, Inc.
358 Veterans Memorial Highway,
Commack, New York 11725

Hauppauge!

Trademarks: IBM PC AT and [illegible] Compaq Computers

Number 122 on the Reader Service Card

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

LOW COST ADVERTISING

If you sell products or services to professional programmers, a ***PJ Billboard*** ad is an easy, low cost way to reach more potential buyers. **Regular listings** include a headline, company info., a Reader Service number and approximateley 400 characters for your message. **Display listings** are 1 5/8" wide by 2 3/8" deep. Costs are $125 for a single insertion, $115 for 3 insertions and $95 for 6 consecutive insertions. ***PJ Billboard*** ads must be prepaid and are non-commissionable. Call the ad department for deadlines.

Programmer's Journal
P.O. Box 30160
Eugene, OR 97403
503-747-0800

COMMUNICATIONS

Custom solutions for Asynchronous, Bisynch, SDLC, Multiline, Multitasking. Unattended Remote Polling Package with Scheduler, Mailbox, Remote Batch and Full Duplex Protocol. Many communications utilities.

ModemArt
P.O. Box 0081
Richardson, TX 75083-0081
214-530-9572 **RS#130**

TURBO PASCAL PROGRAMMERS

The SOURCE CODE for TURBO Pascal is now available. For a remarkably low price users can now have fully commented source code for TURBO Pascal v3.01A (binary arithmetic). This includes the run-time library editor and compiler. Everything you need is included, even an assembler! Free information.

Price: IBM-PC $67.50, Z-80 $45 (specify disk required)

C.C. Software
1907 Alvarado Ave. #B
Walnut Creek, CA 94596
415-939-8153 **RS# 132**

CONVERT BASIC TO C OR PASCAL

BAS_C ™ converts BASICA to C which can be directly compiled by any C compiler. BAS_PAS™ converts BASICA to Turbo Pascal™ Both BAS_C™ and BAS_PAS™ transform the spaghetti code to the scoped, indented structured program automatically. Deadcode. Graphic, Random file.

Sample Conversion: Send your own BASIC program (1 line / 1¢) and receive the printout.

Demo disk $5.00

Gotoless Conversion
P.O. Box 835910
Richardson, TX 75083
214-404-1404 **RS#127**

DRUMA FORTH-83 SYSTEM LANGUAGE

Strict '**83 STANDARD**. Developed for in house use. No 64K limitation. 64K speed & compactness to 320K.

* **IBM PC/XT/AT** & compatibles, **DOS 2.0+**
* **ROM**: Headerless code, separated variables
* Editor, assembler, file & DOS interfaces
* Many powerful and innovative features
* Full Pkg. **$79**, Demo **$10**, S&H **$2**, **VISA/MC**

Druma Forth-83, Druma Inc.
P.O. Box 610097
Austin, TX 78761
512-323-0403 Orders
521-323-2402 BBoard **RS#172**

AUTOMATIC BACKUP

BU.COM is simple to use and effective. Backs up files on an interval you set. For IBM & Compatibles. Not Copy Protected.

Introductory offer—includes **FileTag**, a file utility that lets you copy, view, rename, print, search, erase or change dates/times/attributes on any of the files that you choose.

Both for $25 postpaid
Send check or M.O.
Or write for free information to

Use-Full Software
739 Ewing
Lima, OH 45801
419-229-1919 **RS# 164**

MODULA-2 COMPILER

Generates native code for LARGE and HUGE memory models. Integrated editor and make. Intelligent linker. Inline assembler. Coroutines, IOTRANSFER. Requires PC or clone with 512K, better with 640K. Distributed as Shareware, registration is $35. Download an evaluation copy from our BBS at 404-497-0931, or order from:

Fitted Software Tools
P.O. Box 956129
Diluth, GA 30136 **RS#138**

dBASE TO C TRANSLATOR

dBx is a software system that will allow the programmer to move a dBASE application to a 'C' environment. It includes: (1) a translator that converts dBASE source code into 'C' source code; and (2) a run-time library written in portable 'C'.

For more info call or write

Desktop A.I.
1720 Post Road East
Westport, CT 06880
203-255-3400 **RS#103**

REGISTER NOW AND SAVE

Over 90 seminars and workshops make SOFTWARE DEVELOPMENT '88 the most comprehensive conference and technical seminar ever held on software development. Intimate classes will showcase the practical knowledge of over 70 noted speakers including Bill Gates, Dick Gabriel, Gary Kildall, Alan Kay, PJ. Plauger, Philippe Kahn, Jack Purdum, and Dan Bricklin. Space in classes must be limited. Register now at $575 to avoid sold-out seminars and late registration fees of $120 (after 2/1). Call for more information or register by phone.

SOFTWARE DEVELOPMENT '88
February 17-19
The Ramada Renaissance Hotel
San Francisco Calif.
Sponsored by *COMPUTER LANGUAGE, AI EXPERT,* and *UNIX REVIEW*
500 Howard St.
San Francisco CA 94105
Bill Rutledge, Conference Registrar
(415) 397-1881 x393.
Visa, MC, AMEX accepted

Number **167** on the Reader Service Card

PJ CODE IS AVAILABLE!

Issue listings can be obtained on disk from:

Programmer's Journal
P.O. Box 30160
Eugene, OR 97403
502-747-0800
See ad on page 74.

Listings are also available on

FIDO/OPUS Network
Blazer Net 152 Node 9
503-343-9724

Mnematics Videotex
P. O. Box 19
Sparkhill, NY 10976-0019
800-322-3633
914-761-4700 in NY

Turbo Source Search
P.O. Box 876
Scituate, MA 02066
617-545-9131

The Pointer

The Memory Resident Help System and Demo Program Generator

Custom fit this flashy, menu driven help system to any new or old program without programming, instantly! With the Demo Program Generator, create LIVE DEMOS for any program. Not just another demo program that runs a picture show. Actually run your applications LIVE through the demo windows with *The Pointer.* *Rush $129.95 U.S. to:*

Stau Enterprises Inc.
3650 Kaneff Cres. Suite 1008
Miss. Ont. Can. L5A 4A1
(416) 270 6673

RS# 134

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

The Software Family

Dept PJ18, 649 Mission Street
San Francisco, CA 94105

(1)800-443-7176

In California or outside U.S.
(415)957-0111

Trust TSF to provide the best value and service!

* Technical advice and support by programmers
* Honest and equitable business practices
* Prompt, flexible service to meet your needs

We accept checks, Visa, MasterCard, American Express and COD. We charge your card only when we ship and do **not** add a surcharge. We provide free UPS delivery on software orders over $100 ($3 delivery charge on orders under $100). We add actual charges for UPS Blue Label or Federal Express rush service. Our COD fee is $2. We allow return privileges on most products. We carry a large inventory and ship promptly, so you receive your shipment within 3 to 6 working days of placing your order. Give us a try!

Object Oriented Programming and Prolog

Smalltalk/V from Digitalk provides a complete interactive environment for exploring both object oriented programming and artificial intelligence programming. Smalltalk is both a programming language and a graphics-oriented environment: it provided the conceptual foundation for design of the Xerox Star and Apple Macintosh user interfaces. The Smalltalk/V implementation also includes a Prolog compiler, support for all standard PC hardware and enough power to develop real applications.

Smalltalk/V (list $99.95) $82
"Goodies" Application Disk (list $49) $43
Multi-Processing, Zoom, DOS Calls, Music, Object Loader, and Forward Chaining Inference Engine
EGA/VGA Color Option (list $49) $43
Communications Pack ($49) $43

Basic C Turbo Pascal dBase Publishing AI

C

Blaise **C Tools/5.0** (list$129) ..... **$95** *New!*
Blaise **Turbo C Tools** for Turbo C (list $129) **$95**
Borland **Turbo C v1.5** (list $99) **$68**
Creative **Vitamin C** Specify compiler (list $225) **$159**
Creative Programming **VC Screen** (list $100) **$79**
Gimpel **C-Terp for Turbo C** (list $139) **$95**
Gimpel **PC-Lint** (list $139) **$95**
Metagraphics **Metawindow/All Language** ($195) ... **$159**
Windows, graphics, mouse support
Metagraphics **Metawindow/Turbo C** ($95) **$79**
Microsoft **C v5** w/Codeview & Quick C (list $450) ..... **$279** *New!*
Microsoft **Quick C** (list $99) ..... **$68** *New!*
Softfocus **"C" BTree & ISAM** w/source ($115) **$105**
Sterling Castle **Blackstar "C" Functions** ($129) ..... **$99** *New!*
Full featured library w/source for both MS & Borland
Tenon **WKS Library** read/write Lotus 1-2-3 files ($89) ... **$79** *New!*
Unipress **Phact** RDBMS ($249) ..... **$199** *New!*
includes report writer, query program & all source

BASIC

Borland **Turbo Basic** (list $99) **$68**
Borland **TB Data Base Toolbox** (list $99) **$68**
Cresent **Quick Pack** (list $69) **$59**
Cresent **QBase** (list $99) **$79**
Microsoft **Quick Basic v4** (list $99) ..... **$68** *New!*
Sterling Castle **Basic Development Tools** ($99) **$79**
B-Tree & windows w/source for MS & Borland basics

TURBO PASCAL

Blaise **Turbo Power Tools v4** (list $129) **$95**
Blaise **Turbo Async Plus v4** (list $129) **$95**
Borland **Turbo Pascal v4** (list $99) **$68**
Turbo Power **Turbo Professional 4.0** w/Src ($99) **$79**

OTHER LANGUAGES AND TOOLS

Aldebaran **Source Print** (list $75) **$59**
Aldebaran **Tree Diagrammer** (list $55) **$42**
Coder's Source **Translisp Plus** ($195) ..... **$99** *Save!*
Digitalk **Smalltalk/V** (list $99) **$82**
MKS **Toolkit** Korn shell, vi, grep, 30 more ($139) **$115**
Periscope **Periscope II** (list $175) **$139**
Sharpe **Master*Key** Disasemble, format, X-ref ($80) **$64**
Texas Instruments **PC Scheme** great AI intro ($99) **$77**

TSF carries hundreds of products from dozens of publishers. Call or write for a FREE catalog or a price quote.

Prices good till February 28, 1988

Number 98 on the Reader Service Card

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

ADVERTISER INDEX

Reader Service#	Advertiser Name	Page Number
152	Alpha Computer Service	54
151	American Software International	25
162	Anji Software, Inc.	6
170	ASCII	9
86	Aspen Systems	25
30	Austin Code Works	70
51	Az-Tech Software, Inc.	57
2	Blaise Computing	3
3	Borland International	C4
132	C.C. Software	82
144	*C Gazette, The*	19
38	C Ware Corporation	19
56	CompuView Products, Inc.	63
*	Creative Programming Consultants	80
124	Custom Real Time Software	61
5	Dair Computer Systems	18
153	Definicon Systems	64
103	Desktop AI	82
172	Druma, Inc.	82
70	E.C.I.	40
8	Ecosoft, Inc.	5
113	Elfsoft	66
138	Fitted Software Tools	82
118	Fleming Software	30
135	Gilmore Systems	32
*	Gimpel Software	48
157	Gold Brick Software	58
127	Gotoless Conversion	82
122	Hauppauge Computer Works, Inc.	81
154	Haven Tree Software LTD	67
57	Hawkeye Grafix Inc.	31
169	IGC	1
139	I.M.S.I.	C3
159	I.M.S.I.	45
155	Komputerwerk	57
63	Laboratory Microsystems, Inc.	21
44	Lahey Computer Systems	57
92	Manx Software Systems	51
141	Merlin Publishing Group	18
143	Metagraphics Software Corporation	68
45	MetaWare, Inc.	75
*	**PLEASE CONTACT THESE ADVERTISERS DIRECTLY**	

Reader Service#	Advertiser Name	Page Number
168	Michael J. Young Software Engineering	57
13	Micromotion	65
104	MicroWay	55
121	MISOSYS Inc.	59
94	MMC AD Systems	15
130	ModemArt	82
109	NanoSoft	15
160	Nu-mega Technologies	35
16	Opt-Tech Data Processing	27
145	Pathfinder Associates	45
80	Paul Mace Software	7
6	Periscope Company, The	79
146	Programmer's Connection	41-43
*	*Programmer's Journal* 5 Year Index	71-74
*	Quaid Software Limited	31
97	Qualstar Corporation	21
147	Quarterdeck Office Systems	C1
46	Quilt Computing	30
*	Raima Corporation	49
156	SAIC	60
165	SLR Systems	77
148	SemWare	78
161	Serengeti Software	69
167	Software Development '88	82
98	Software Family, The	83
115	Software Interphase, Inc.	14
110	Softway, Inc.	27
47	Solution Systems	23
134	Stau Enterprises Inc.	82
158	Summit Information Systems, Inc.	56
85	Sunny Hill Software	59
74	Sydetech, Inc.	11
171	Systemat Corporation	10
163	Tenon Software	21
117	Tom Rettig Associates	65
166	Tool Makers, The	65
35	Turbo Source Search	40
164	Use-Full Software	82
125	V-Communications	25
73	Whitewater Group, The	8
123	Xenosoft	27
*	**PLEASE CONTACT THESE ADVERTISERS DIRECTLY**	

When contacting any of the above advertisers, please tell them you saw their ad in ***Programmer's Journal*** !

Articles Coming in the March/April Issue 6.2

PJ Forum:

RISCs, CISCs and FABs
by Hal W. Hardenbergh

On Graphics:

Yet Another Write Mode
by Michael Abrash

Pascal:

Turbo Pascal 4.0 Reviewed
by John J. Newlin

ADVERTISE IN

Programmer's Journal
March/April issue 6.2
New Space Reservation Deadline:
January 8, 1988
Materials Due:
January 22, 1988
Call the ***Programmer's Journal***
Advertising Department to reserve your space!

(503) 747-0800

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

TURBO PASCAL AND TURBO C... MEET TURBOHALO

"Ideal! TurboHALO does the job comparable to packages costing $3000 to $4000."
Jim Bromley
Superintendent of Spectrum Management

"TurboHALO is so fun... I use it to design programs as a hobby...It's got lots of ability."
William Porter
Control Systems Manager

"I like the speed of TurboHALO...it's ten times faster than the competition."
Deniz Terry
Doctoral Candidate

"We evaluated all of the graphic development packages for Turbo Pascal, and TurboHALO was the hands down winner!"
Quinn Curtis
Largest New England Distributor

It's time to put graphics into your programming.
A picture's worth a thousand words. So your programming isn't complete until you have graphics. TurboHALO brings your screen and printer to life with subroutines that draw, chart, map, and display. All with color, shape, clarity, perspective and motion. With TurboHALO, create any picture you can imagine.

TurboHALO gives you graphics power.
TurboHALO gives you everything you need for Turbo C and Turbo Pascal graphics programming. A library of over 150 graphics subroutines. Drivers for over 42 graphics hardware devices for the IBM PC family and compatibles. You can create the images you want, on the hardware you have!

Fast, proven and reliable.
TurboHALO is up to ten times faster than other graphics toolkits. And with TurboHALO you get proven, reliable programming tools used by professionals for years.

You'll like TurboHALO or your money back!
TurboHALO is available for only $95. You get an unconditional 45-day money-back guarantee on TurboHALO. To order, call your dealer or IMSI at (415) 454-7101 or (800) 222-4723; in CA (800) 562-4723; in Washington DC (202) 363-9340 or in NC (919) 854-4674.

YES, *I want to see the difference TurboHALO makes in my graphics programming! Rush me the following TurboHALO Graphics Toolkit(s) @ $95 each plus $3 shipping. California residents add 6% sales tax.*

☐ *TurboHALO for Turbo Pascal*
☐ *TurboHALO for Turbo C*
☐ *Check enclosed for $ _____ (made payable to IMSI)*
☐ *Charge my credit card for $ _____* ☐ *VISA* ☐ *MasterCard*

Signature _____
Card Number _____ *Expiration Date*
Name _____
Title _____
Company _____
Street _____
City _____ *State* _____ *Zip*

IMSI™

Mail to:
IMSI, 1299 Fourth Street, San Rafael, CA 94901

PJ188

FOR GRAPHICS PROGRAMMING

TurboHALO requires 256K memory (min); memory resident drivers require 2K (Turbo Pascal only); DOS version 2.0 or higher; Borland language/compiler required. TurboHALO is a trademark of Media Cybernetics and IMSI. Turbo C and Turbo Pascal are trademarks of Borland.

Number **139** on the Reader Service Card

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED

Program in the fast lane with Borland's new Turbo Pascal 4.0!

Our new Turbo Pascal® 4.0 is so fast, it's almost reckless. How fast? Better than 27,000 lines of code per minute. That's more than twice as fast as 3.0 and the reason why you need 4.0 today.

4.0 breaks the code barrier

No more swapping code in and out to beat the 64K code barrier. Designed for large programs, Turbo Pascal 4.0 lets you use every byte of memory in your computer.

4.0 uses logical units for separate compilation

Pascal 4.0 lets you break up the code gang into "units," or "chunks." These logical modules can be worked with swiftly and separately. Compiling and linking these separate units happens in a flash because your compiling horsepower is better than 27,000 lines a minute.* And 4.0 also includes an automatic project Make.

4.0 Highlights:

- Compiles 27,000 lines per minute
- Includes automatic project Make
- Supports > 64K programs
- Uses units for separate compilation
- Integrated development environment
- Interactive error detection/location
- Includes a command-line version of the compiler

60-Day Money-back Guarantee**

*Run on an 8MHz IBM AT.

**If within 60 days of purchase this product does not perform in accordance with our claims, call our customer service department, and we will arrange a refund.

All Borland products are trademarks or registered trademarks of Borland International, Inc. Copyright ©1987 Borland International, Inc. BI 1161B

4.0's cursor automatically lands on any trouble spot

4.0's interactive error detection and location means that the cursor automatically lands where the error is. While you're compiling or running a program, you get an error message *and* the cursor flags the error's location for you.

4.0 gives you an integrated programming environment

4.0's integrated environment includes pull-down menus and a built-in editor. Your program output is automatically saved and shown in the output window. You can Scroll, Pan, or Page through all your output and know where everything is all the time. Given 4.0's integration, you can edit, compile, find and correct errors—all from inside the integrated development environment. We even include a command line version of the compiler.

Compatibility with Turbo Pascal 3.0

We've created Version 4.0 to be highly compatible with Version 3.0 and included a conversion program and compatibility units to help you convert 3.0 programs to 4.0.

4.0 is all yours for only $99.95

Sieve (25 iterations)

	Turbo Pascal 4.0	Turbo Pascal 3.0
Size of Executable File	2224 bytes	11682 bytes
Execution speed	9.3 seconds	9.7 seconds

Sieve of Eratosthenes, run on an 8MHz IBM AT

Since the source file above is too small to indicate a difference in compilation speed we compiled our CHESS program from Turbo Gameworks to give you a true sense of how much faster 4.0 really is!

Compilation of CHESS.PAS (5469 lines)

	Turbo Pascal 4.0	Turbo Pascal 3.0
Compilation speed	12.1 seconds	35.5 seconds
Lines per minute	27,119	9,243

CHESS.PAS compiled on an 8 MHz IBM AT

For the IBM PS/2™ and the IBM® and Compaq® families of personal computers and all 100% compatibles

For the dealer nearest you,or to order now,

Call (800) 543-7543

BORLAND INTERNATIONAL

BORLAND

YES! I want to upgrade to Turbo Pascal 4.0 and the 4.0 Toolboxes

Registered owners have been notified by mail. If you are a registered Turbo Pascal user and have not been notified of Version 4.0 by mail, please call us at (800) 543-7543. To upgrade if you have not registered your product, just send the original registration form from your manual and payment with this completed coupon to:

Pascal 4.0 Upgrade Dept.
Borland International
4585 Scotts Valley Drive
Scotts Valley, CA 95066

Name _____

Ship Address _____

City _____ State _____

Zip _____ Telephone ( ) _____

Please check box(es)	Suggested Retail	Upgrade†	Serial No.
☐ Turbo Pascal 4.0 Compiler	$ 99.95	$ 39.95	_____
☐ Turbo Pascal Tutor	69.95	19.95	_____
☐ Turbo Pascal Database Toolbox	99.95	29.95	_____
☐ Turbo Pascal Graphix Toolbox	99.95	29.95	_____
☐ Turbo Pascal Editor Toolbox	99.95	29.95	_____
☐ Turbo Pascal Numerical Methods Toolbox	99.95	29.95	_____
☐ Turbo Pascal Gameworks	99.95	29.95	_____

Total product amount $ _____
CA and MA residents add sales tax $ _____
In US please add $5 shipping and handling for each product $ _____
Outside US please add $10 shipping and handling for each product $ _____
Total amount enclosed $ _____

Please specify diskette size ☐ 5¼" ☐ 3½"

Payment: ☐ VISA ☐ MC ☐ Check ☐ Bank Draft

Credit card expiration date: _____/_____

Card # _____

†To qualify for the upgrade price you must give the serial number of the equivalent product you are upgrading.

For the IBM PS/2™ and the IBM® and Compaq® families of personal computers and all 100% compatibles.

This offer is limited to one upgrade per valid registered product. It is good until June 30, 1988. Not good with any other offer from Borland. Please allow 4 to 6 weeks for delivery of Toolboxes.

Outside U.S. make payments by bank draft payable in U.S. dollars drawn on a U.S. bank. CODs and purchase orders will not be accepted by Borland.

Number 3 on the Reader Service Card

PJ 1/2 88

PRODUCED 2004 BY UNZ.ORG
ELECTRONIC REPRODUCTION PROHIBITED